	Type	Hits	Search Text	DBs
1	BRS	28	6457103.pn. or 6507891.pn. or 6574715.pn. or 6557076.pn. or 6615235.pn. or 6584548.pn. or 4,638,431.pn. or 5,025365.pn. or 5,136,696.pn. or 5,371,896.pn. or 5,511,212.pn. or 5119,862.pn. or 5,524,190.pn. or 5,551,000.pn. or 5,598,551.pn. or 5,625,789.pn. or 5,706,435.pn. or 5,717,897.pn. or 5,745,728.pn. or 5,819,301.pn. or 5,860,110.pn. or 5,961,602.pn. or 5,974,421.pn. or 6,000,012.pn. or 6,014,698.pn. or 6,021,426.pn. or 6,038,601.pn. or 6,052,698.pn.	USPAT; US-PGPUB; EPO; JPO; IBM_TDB

	Time Stamp
1	2004/08/02 10:26

	Type	Hits	Search Text	DBs
2	BRS	19	(6457103.pn. or 6507891.pn. or 6574715.pn. or 6557076.pn. or 6615235.pn. or 6584548.pn. or 4,638,431.pn. or 5,025365.pn. or 5,136,696.pn. or 5,371,896.pn. or 5,511,212.pn. or 5119,862.pn. or 5,524,190.pn. or 5,551,000.pn. or 5,598,551.pn. or 5,625,789.pn. or 5,706,435.pn. or 5,717,897.pn. or 5,745,728.pn. or 5,819,301.pn. or 5,860,110.pn. or 5,961,602.pn. or 5,974,421.pn. or 6,000,012.pn. or 6,014,698.pn. or 6,021,426.pn. or 6,038,601.pn. or 6,052,698.pn.) and (cache or buffer) with (instruction or command)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB

	Time Stamp
2	2004/08/02 10:34

	Type	Hits	Search Text	DBs
3	BRS	5	(6457103.pn. or 6507891.pn. or 6574715.pn. or 6557076.pn. or 6615235.pn. or 6584548.pn. or 4,638,431.pn. or 5,025365.pn. or 5,136,696.pn. or 5,371,896.pn. or 5,511,212.pn. or 5119,862.pn. or 5,524,190.pn. or 5,551,000.pn. or 5,598,551.pn. or 5,625,789.pn. or 5,706,435.pn. or 5,717,897.pn. or 5,745,728.pn. or 5,819,301.pn. or 5,860,110.pn. or 5,961,602.pn. or 5,974,421.pn. or 6,000,012.pn. or 6,014,698.pn. or 6,021,426.pn. or 6,038,601.pn. or 6,052,698.pn.) and server with (primary or secondary instruction or command)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB

	Time Stamp
3	2004/08/02 11:06

	Type	Hits	Search Text	DBs
4	BRS	0	(6457103.pn. or 6507891.pn. or 6574715.pn. or 6557076.pn. or 6615235.pn. or 6584548.pn. or 4,638,431.pn. or 5,025365.pn. or 5,136,696.pn. or 5,371,896.pn. or 5,511,212.pn. or 5119,862.pn. or 5,524,190.pn. or 5,551,000.pn. or 5,598,551.pn. or 5,625,789.pn. or 5,706,435.pn. or 5,717,897.pn. or 5,745,728.pn. or 5,819,301.pn. or 5,860,110.pn. or 5,961,602.pn. or 5,974,421.pn. or 6,000,012.pn. or 6,014,698.pn. or 6,021,426.pn. or 6,038,601.pn. or 6,052,698.pn.) and cofetch$4	USPAT; US-PGPUB; EPO; JPO; IBM_TDB

	Time Stamp
4	2004/08/02 11:06

	Type	Hits	Search Text	DBs
5	BRS	5	(6457103.pn. or 6507891.pn. or 6574715.pn. or 6557076.pn. or 6615235.pn. or 6584548.pn. or 4,638,431.pn. or 5,025365.pn. or 5,136,696.pn. or 5,371,896.pn. or 5,511,212.pn. or 5119,862.pn. or 5,524,190.pn. or 5,551,000.pn. or 5,598,551.pn. or 5,625,789.pn. or 5,706,435.pn. or 5,717,897.pn. or 5,745,728.pn. or 5,819,301.pn. or 5,860,110.pn. or 5,961,602.pn. or 5,974,421.pn. or 6,000,012.pn. or 6,014,698.pn. or 6,021,426.pn. or 6,038,601.pn. or 6,052,698.pn.) and (fetch$4 adj5 (instruct$4 or command))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB

	Time Stamp
5	2004/08/02 11:08

	Type	Hits	Search Text	DBs
6	BRS	1	(6457103.pn. or 6507891.pn. or 6574715.pn. or 6557076.pn. or 6615235.pn. or 6584548.pn. or 4,638,431.pn. or 5,025365.pn. or 5,136,696.pn. or 5,371,896.pn. or 5,511,212.pn. or 5119,862.pn. or 5,524,190.pn. or 5,551,000.pn. or 5,598,551.pn. or 5,625,789.pn. or 5,706,435.pn. or 5,717,897.pn. or 5,745,728.pn. or 5,819,301.pn. or 5,860,110.pn. or 5,961,602.pn. or 5,974,421.pn. or 6,000,012.pn. or 6,014,698.pn. or 6,021,426.pn. or 6,038,601.pn. or 6,052,698.pn.) and (prefetch$4 adj5 (instruct$4 or command))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB
7	BRS	7234	(web or server) with cache or cluster adj3 cache	USPAT; US-PGPUB; EPO; JPO; IBM_TDB
8	BRS	1318	(cofetch$4 or prefetch$4) with (multiple or plural$4 or more or several or second) with (command or instruction)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB
9	BRS	11	((cofetch$4 or prefetch$4) with (multiple or plural$4 or more or several or second) with (command or instruction)) same ((web or server) with cache or cluster adj3 cache)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB
10	BRS	1	6282602.pn.	USPAT; US-PGPUB; EPO; JPO; IBM_TDB

	Time Stamp
6	2004/08/02 11:17
7	2004/08/02 15:19
8	2004/08/02 11:41
9	2004/08/02 11:22
10	2004/08/02 11:25

	Type	Hits	Search Text	DBs
11	BRS	8707	((command or instruction) and single).ab.	USPAT; US-PGPUB; EPO; JPO; IBM_TDB
12	BRS	498	(jack and lane).xa. or (jack and lane).xp.	USPAT; US-PGPUB; EPO; JPO; IBM_TDB
13	BRS	8	(((command or instruction) and single).ab.) and ((jack and lane).xa. or (jack and lane).xp.)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB
14	BRS	19581	((command or instruction) and two).ab.	USPAT; US-PGPUB; EPO; JPO; IBM_TDB
15	BRS	13	((jack and lane).xa. or (jack and lane).xp.) and (((command or instruction) and two).ab.)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB
16	BRS	8	(((jack and lane).xa. or (jack and lane).xp.) and (((command or instruction) and two).ab.)) not ((((command or instruction) and single).ab.) and ((jack and lane).xa. or (jack and lane).xp.))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB
17	BRS	1980	pre adj2 execut$4 or post adj2 execut$4 or preexecut$4 or postexecut$4	USPAT; US-PGPUB; EPO; JPO; IBM_TDB
18	BRS	55	(pre adj2 execut$4 or post adj2 execut$4 or preexecut$4 or postexecut$4) and ((web or server) with cache or cluster adj3 cache)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB
19	BRS	1	(pre adj2 execut$4 or post adj2 execut$4 or preexecut$4 or postexecut$4) same ((web or server) with cache or cluster adj3 cache)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB

	Time Stamp
11	2004/08/02 11:29
12	2004/08/02 11:26
13	2004/08/02 11:26
14	2004/08/02 11:29
15	2004/08/02 11:30
16	2004/08/02 11:35
17	2004/08/02 11:36
18	2004/08/02 11:37
19	2004/08/02 11:36

	Type	Hits	Search Text	DBs
20	BRS	67	(pre adj2 execut$4 or post adj2 execut$4 or preexecut$4 or postexecut$4) and ((cofetch$4 or prefetch$4) with (multiple or plural$4 or more or several or second) with (command or instruction))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB
21	BRS	4	(pre adj2 execut$4 or post adj2 execut$4 or preexecut$4 or postexecut$4) same ((cofetch$4 or prefetch$4) with (multiple or plural$4 or more or several or second) with (command or instruction))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB
22	BRS	914	(cofetch$4 or prefetch$4) with (multiple or plural$4 or more or several or second) with (command or instruction) same (cache or buffer$4 or caching)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB
23	BRS	736	(cofetch$4 or prefetch$4) with (multiple or plural$4 or more or several or second) with (command or instruction) with (cache or buffer$4 or caching)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB
24	BRS	6	((cofetch$4 or prefetch$4) with (multiple or plural$4 or more or several or second) with (command or instruction) with (cache or buffer$4 or caching)) and (jsp or java adj3 server adj3 page)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB
25	BRS	1	(cofetch$4) with (multiple or plural$4 or more or several or second) with (command or instruction) with (cache or buffer$4 or caching)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB

	Time Stamp
20	2004/08/02 11:37
21	2004/08/02 11:37
22	2004/08/02 11:42
23	2004/08/02 14:22
24	2004/08/02 14:21
25	2004/08/02 14:22

	Type	Hits	Search Text	DBs
26	BRS	8	(cofetch$4) same (multiple or plural$4 or more or several or second) with (command or instruction) with (cache or buffer$4 or caching)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB
27	BRS	1	6584548.pn.	USPAT; US-PGPUB; EPO; JPO; IBM_TDB
28	BRS	5	"20020143868" or 5,822,749.pn. or 6,298,356.pn. or 6,282,548.pn or 6,643,652.pn.or "20020010753"	USPAT; US-PGPUB; EPO; JPO; IBM_TDB
29	BRS	2	((web or server) and (cache or buffer) and (command or instruct$4)).ti.	USPAT; US-PGPUB; EPO; JPO; IBM_TDB

	Time Stamp
26	2004/08/02 14:59
27	2004/08/02 15:06
28	2004/08/02 15:18
29	2004/08/02 15:20

Access DB# 128748

SEARCH REQUEST FORM

(3)

Scientific and Technical Information Center

Requester's Full Name: Jack Lane Examiner # : 68699 Date: 08/02/04
Art Unit: 2188 Phone Number 305-3818 Serial Number: 09/740,399
Mail Box Location: 2Y13 Results Format Preferred (circle): PAPER DISK E-MAIL

If more than one search is submitted, please prioritize searches in order of need.

****

Please provide a detailed statement of the search topic, and describe as specifically as possible the subject matter to be searched. Include the elected species or structures, keywords, synonyms, acronyms, and registry numbers, and combine with the concept or utility of the invention. Define any terms that may have a special meaning. Give examples or relevant citations, authors, etc, if known. Please attach a copy of the cover sheet, pertinent claims, and abstract.

Title of Invention: Information processing apparatus, method thereof, information processing system, and medium

Inventors (please provide full names): see Bib sheet

Earliest Priority Filing Date: 12/18/2000

***For Sequence Searches Only* Please include all pertinent information (parent, child, divisional, or issued patent numbers) along with the appropriate serial number.**

See abstract and claims. Invention essentially cofetches/prefetches instructions and stores them in a web cache

Search:
Web or server () cache or cluster cache
Cofetch$4 or prefetch$4 () command or instruction
Pre-execute
Post execute

****

STAFF USE ONLY

Searcher: Terese Esterhold
Searcher Phone #: 308-7795
Searcher Location: 4B30
Date Searcher Picked Up: 8/4/04 8:45 am
Date Completed: 8/6/04 10:16 am
Searcher Prep & Review Time:
Clerical Prep Time:

Type of Search

NA Sequence (#)
AA Sequence (#)
Structure (#)
Bibliographic
Litigation
Fulltext
Patent Family

Vendors and cost where applicable

STN
Dialog
Questel/Orbit
Dr.Link
Lexis/Nexis
Sequence Systems
WWW/Internet

```
Set     Items   Description
S1       1007   (COFETCH??? OR (CO OR PRE)()FETCH??? OR PREFETCH?)(3N)(COM-
            MAND? OR INSTRUCTION?)
S2       2452   (PRIMARY OR PRIME OR FIRST OR 1ST OR INITIAL OR LEADING OR
            MAIN OR DOMINANT OR CARDINAL OR ORIGINAL)()(COMMAND? OR INSTR-
            UCTION?)
S3    3373668   EXECUT? OR PERFORM? OR CONDUCT? OR MANAG? OR ACCOMPLISH?
S4        334   (PRE OR PRIOR OR BEFORE?)()(S3 (3N)(COMMAND? OR INSTRUCTIO-
            N?))
S5       3406   (SECONDARY OR CHILD OR BRANCH OR LEAVES OR OFFSPRING OR OF-
            F()SPRING OR SUBPROGRAM OR SUBROUTINE OR SLAVE OR CALLED)(2W)-
            (COMMAND? OR INSTRUCTION?)
S6       2115   (SEND? OR TRANSMIT? OR TRANSFER? OR CONVEY? OR DELIVER? OR
            OUTPUT? OR RETURN? OR DISPATCH?)(2W)((COMPUTER? OR CLIENT)(2N-
            )(NODE? OR PC OR PERSONAL()COMPUTER? OR WORKSTATION? OR WORK(-
            )STATION?))
S7        631   (POST OR AFTER OR SUBSEQUENT? OR LATER OR FOLLOWING)()(S3 -
            (3N) (COMMAND? OR INSTRUCTION))
S8        302   (STORING OR STORE? OR SAVE OR SAVING OR KEEP OR KEEPING OR
            PRESERV?)(3N) S5
S9    1829010   CACHE OR BUFFER? OR STORAGE OR MEMORY OR REPOSITOR?
S10        34   S1 AND S2 AND S3
S11        26   S1 AND S4 AND S9
S12        11   S7 AND S3 AND S2 AND S5
S13        65   S10 OR S11 OR S12
S14        64   S13 AND IC=G06F?
S15         8   S14 AND IC=G06F-015?
S16        10   S13 AND MC=(T01-F03A OR T01-F05A OR T01-F07 OR T01-H03A OR
            T01-N01A2A OR T01-N02A2C OR T01-N03B1 OR T01-S03)
S17        16   S15 OR S16
File 347:JAPIO Nov 1976-2004/Apr(Updated 040802)
         (c) 2004 JPO & JAPIO
File 350:Derwent WPIX 1963-2004/UD,UM &UP=200449
         (c) 2004  Thomson Derwent
```

17/5/1 (Item 1 from file: 347)
DIALOG(R)File 347:JAPIO
(c) 2004 JPO & JAPIO. All rts. reserv.

06083428 **Image available**
MICROCOMPUTER

PUB. NO.: 11-024942 [JP 11024942 A]
PUBLISHED: January 29, 1999 (19990129)
INVENTOR(s): UENO TOMOO
APPLICANT(s): NEC KYUSHU LTD
APPL. NO.: 09-184172 [JP 97184172]
FILED: July 09, 1997 (19970709)
INTL CLASS: **G06F-009/46 ; G06F-009/46 ; G06F-009/46 ; G06F-015/78**

ABSTRACT

PROBLEM TO BE SOLVED: To start **execution** of an instruction without waiting for the storage of an instruction code in an **instruction** queue with a **prefetch** operation and to improve the response speed of restoration to a main routine from an interruption sub-routine.

SOLUTION: A **first instruction** queue 1 to become operable at the time of a main routine processing and a second instruction queue 2 to become operable at the time of an interruption sub-routine processing are provided. An interruption controller 10 is operated by switching the **first instruction** queue 1 or the second instruction queue 2 according to whether the present processing is the main routine processing or the interruption sub-routine processing.

COPYRIGHT: (C)1999,JPO

17/5/2 (Item 2 from file: 347)
DIALOG(R)File 347:JAPIO
(c) 2004 JPO & JAPIO. All rts. reserv.

02962975 **Image available**
INFORMATION PROCESSOR

PUB. NO.: 01-260575 [JP 1260575 A]
PUBLISHED: October 17, 1989 (19891017)
INVENTOR(s): YAMAMOURI MASAHIKO
APPLICANT(s): NEC CORP [000423] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 63-089936 [JP 8889936]
FILED: April 11, 1988 (19880411)
INTL CLASS: [4] **G06F-015/347 ; G06F-009/38**
JAPIO CLASS: 45.4 (INFORMATION PROCESSING -- Computer Applications); 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units)
JOURNAL: Section: P, Section No. 989, Vol. 14, No. 15, Pg. 8, January 12, 1990 (19900112)

ABSTRACT

PURPOSE: To reduce overhead and to improve performance by decoding the content of an **instruction buffer before performing** a decode processing at the decoding stage of an **instruction** , starting the **pre - fetch** of a mask vector, and taking out the mask vectors corresponding to the number of vector elements.

CONSTITUTION: The title device is provided with the instruction **buffer** 1, an instruction register 2, universal register 3, address adder 4, instruction decoder 5, address adder 7, and a mask vector **buffer** 8. And the content of the instruction **buffer** 1 is decoded before the decode processing at the decoding stage of the instruction is performed, and the pre-fetch of the mask vector is started, and the mask vectors corresponding to the number of vector elements are taken out. In such a way, it is possible to reduce the overhead at the time of fetching the mask vector in

a vector instruction processing with a masking function, and to suppress the fetch of an unrequired mask vector, and to improve the performance.

17/5/3 (Item 3 from file: 347)
DIALOG(R)File 347:JAPIO
(c) 2004 JPO & JAPIO. All rts. reserv.

02962974 **Image available**
INFORMATION PROCESSOR

PUB. NO.: 01-260574 [JP 1260574 A]
PUBLISHED: October 17, 1989 (19891017)
INVENTOR(s): YAMAMOURI MASAHIKO
APPLICANT(s): NEC CORP [000423] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 63-089935 [JP 8889935]
FILED: April 11, 1988 (19880411)
INTL CLASS: [4] **G06F-015/347 ; G06F-009/38**
JAPIO CLASS: 45.4 (INFORMATION PROCESSING -- Computer Applications); 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units)
JOURNAL: Section: P, Section No. 989, Vol. 14, No. 15, Pg. 8, January 12, 1990 (19900112)

ABSTRACT

PURPOSE: To reduce overhead in the fetch of a mask vector in a vector instruction processing with a masking function by decoding the content of an **instruction buffer before performing** a decode processing at the decoding stage of an **instruction** , and starting the **pre - fetch** of the mask vector.

CONSTITUTION: The title device is provided with the instruction **buffer** 1, an instruction register 2, universal register 3, address adder 4, selector 5, instruction decoder 6, address adder 7, mask vector **buffer** 8, and a pre-fetch control circuit 9. And the content of the instruction **buffer** 1 is decoded before the decode processing at the decoding stage of the instruction is performed, and the pre- fetch of the mask vector is started. In such a way, it is possible to reduce the overhead in the fetch of the mask vector in the vector instruction processing with the masking function.

17/5/4 (Item 1 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

014938623 **Image available**
WPI Acc No: 2002-759332/200282
XRPX Acc No: N02-597886
Distributed web application supporting software system for Internet, has pre - execution **and** post - execution instruction **sequences for** executing **primary and** secondary commands **and storing** executed **commands in** cache
Patent Assignee: CONNER M H (CONN-I); COPELAND G P (COPE-I); FLURRY G A (FLUR-I)
Inventor: CONNER M H; COPELAND G P; FLURRY G A
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 20020116448	A1	20020822	US 2000740399	A	20001218	200282	B

Priority Applications (No Type Date): US 2000740399 A 20001218
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 20020116448	A1		16	G06F-015/16	

Abstract (Basic): US 20020116448 A1
NOVELTY - The system has a **pre - execution instruction** sequence

which is invoked after the **execution** of a **primary command** in order to **execute** several **secondary commands** and return all the **executed** commands to the client. A **post - execution instruction** sequence is invoked after the **execution** of the **primary command** to store the **secondary commands** in a **cache** along with the **primary command** .

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are included for the following:

(1) Method of **cofetching** several **commands** in an object oriented software system;

(2) **commands cofetching** system;

(3) computer program product for **cofetching commands** ; and

(4) Computer product comprising the software system.

USE - For supporting distributed web applications between client and server connected to network such as Internet.

ADVANTAGE - Improves server responsiveness by avoiding the need to issue separate request for every command. Avoids the need for accurate anticipation of the server-to-command ratio and thus optimizes the use of the **cache** .

DESCRIPTION OF DRAWING(S) - The figure shows the fragment and data granularity for a web site.

pp; 16 DwgNo 2/5

Title Terms: DISTRIBUTE; WEB; APPLY; SUPPORT; SOFTWARE; SYSTEM; PRE; **EXECUTE** ; POST; **EXECUTE** ; INSTRUCTION; SEQUENCE; **EXECUTE** ; PRIMARY; SECONDARY; COMMAND; **STORAGE** ; **EXECUTE** ; COMMAND; **CACHE**

Derwent Class: T01

International Patent Class (Main): **G06F-015/16**

File Segment: EPI

17/5/5 (Item 2 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2004 Thomson Derwent. All rts. reserv.

014178161 **Image available**

WPI Acc No: 2001-662389/200176

XRPX Acc No: N01-493468

Instruction pre - fetching method in computer system, involves accessing target block of target instruction to determine predicted trace value and executing **pre - fetch instruction only when predicted value is greater**

Patent Assignee: HEWLETT-PACKARD CO (HEWP)

Inventor: GORNISH E H

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 6314431	B1	20011106	US 99387392	A	19990902	200176	B

Priority Applications (No Type Date): US 99387392 A 19990902

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 6314431	B1		24	G06F-017/30	

Abstract (Basic): US 6314431 B1

NOVELTY - A target basic block of a target instruction is accessed to determine a predicted trace value which is then compared with a minimum value. A cost-effective, **pre - fetch instruction** is executed only when the predicted trace value is greater. The next branch instruction is accessed and labeled depending on the execution status.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:

(a) **Instruction pre - fetching** tool;

(b) Computer readable medium storing pre-fetching program

USE - For cost-effective **pre - fetching** of **instruction** , data in **cache** in multi-purpose computer system.

ADVANTAGE - Since the **pre - fetch instruction** having higher

probability of usage are executed, useless pre-fetch is avoided, hence pre-fetch is cost-effective. Also, coverage of **pre - fetched instructions** is improved.

DESCRIPTION OF DRAWING(S) - The figure shows a flowchart of operation of pre-fetching tool.

pp; 24 DwgNo 5A/5

Title Terms: INSTRUCTION; PRE; FETCH; METHOD; COMPUTER; SYSTEM; ACCESS; TARGET; BLOCK; TARGET; INSTRUCTION; DETERMINE; PREDICT; TRACE; VALUE; EXECUTE; PRE; FETCH; INSTRUCTION; PREDICT; VALUE; GREATER

Derwent Class: T01

International Patent Class (Main): **G06F-017/30**

International Patent Class (Additional): **G06F-012/00** ; **G06F-015/00**

File Segment: EPI

17/5/6 (Item 3 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2004 Thomson Derwent. All rts. reserv.

014056183 **Image available**

WPI Acc No: 2001-540396/200160

XRPX Acc No: N01-401528

Data pre - fetch **method for** instruction **fetch from** memory , **involves pre-fetch history table with associated validity bits**

Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)

Inventor: BURKY W E; LENK P S; NGUYEN D Q; SCHROTER D A; TUNG S S; VADEN M T

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 6275918	B1	20010814	US 99268307	A	19990316	200160	B

Priority Applications (No Type Date): US 99268307 A 19990316

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 6275918	B1		9	G06F-009/34	

Abstract (Basic): US 6275918 B1

NOVELTY - The pre-fetch method uses a pre-fetch history table (52) to compare parts of an instruction address to address entries in the table based on the status of a entry validity bit. If the bit is set and the addresses match, an indicator field in the entry is checked with reference to a threshold level. If the indicator is greater than the level, a target operand address (70) is pre-fetched based on stride and direction.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:

(1) An information handling system using a pre-fetch history table.

(2) A computer program for **executing pre - fetch instructions** within a **pre - fetch** history table architecture.

USE - For use in computer processor architectures with **instruction pre - fetch** capabilities.

ADVANTAGE - The method provides improved pre-fetch accuracy in a data processing system by using a pre-fetch table (claimed).

DESCRIPTION OF DRAWING(S) - The block diagram represents a data pre-fetch system using a pre-fetch history table.

Pre-fetch history table (52)

Target operand address (70)

pp; 9 DwgNo 2/3

Title Terms: DATA; PRE; FETCH; METHOD; INSTRUCTION; FETCH; **MEMORY** ; PRE; FETCH; HISTORY; TABLE; ASSOCIATE; VALID; BIT

Derwent Class: T01

International Patent Class (Main): G06F-009/34

File Segment: EPI

17/5/7 (Item 4 from file: 350)

DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

012365417 **Image available**
WPI Acc No: 1999-171524/199915
XRPX Acc No: N99-125547

Microcomputer with interruption processing function - has instruction controller which switches over either first or second instruction queues that can be operated during main routine process or subroutine process of predetermined interruption, respectively.

Patent Assignee: NEC KYUSHU LTD (KYUN)
Number of Countries: 001 Number of Patents: 002
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
JP 11024942	A	19990129	JP 97184172	A	19970709	199915	B
JP 3123946	B2	20010115	JP 97184172	A	19970709	200106	

Priority Applications (No Type Date): JP 97184172 A 19970709
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
JP 11024942	A		9	G06F-009/46	
JP 3123946	B2		9	G06F-009/46	Previous Publ. patent JP 11024942

Abstract (Basic): JP 11024942 A

NOVELTY - An instruction controller (10) switches over either a **first instruction** queue (1), that can be operated during a main routine process, or a second instruction queue (2) that can be operated during the subroutine process of a predetermined interruption.

USE - None given.

ADVANTAGE - Improves response speed of restoration from an interruption subroutine to a main routine since start instruction **execution** of instruction code is enabled. Instruction code can be started without waiting for the storage of the instruction code in an **instruction** queue by **pre - fetch** operation. DESCRIPTION OF DRAWING(S) - The figure shows the block diagram of a microcomputer. (1,2) Instruction queues; (10) Instruction controller.

Dwg.1/14

Title Terms: MICROCOMPUTER; INTERRUPT; PROCESS; FUNCTION; INSTRUCTION; CONTROL; SWITCH; FIRST; SECOND; INSTRUCTION; QUEUE; CAN; OPERATE; MAIN; ROUTINE; PROCESS; SUBROUTINES; PROCESS; PREDETERMINED; INTERRUPT; RESPECTIVE
Derwent Class: T01
International Patent Class (Main): **G06F-009/46**
International Patent Class (Additional): **G06F-009/38 ; G06F-015/78**
File Segment: EPI

17/5/8 (Item 5 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

011582682 **Image available**
WPI Acc No: 1997-559163/199751
XRPX Acc No: N97-465987

Asynchronous transfer mode cell processing method - executing subsequent branch instruction **based on modulo portion of result of** first instruction **if** first instruction **include arithmetic operation indication**

Patent Assignee: MAKER COMMUNICATIONS INC (MAKE-N)
Inventor: BERGANTINO P V; LUSSIER D J
Number of Countries: 069 Number of Patents: 003
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
WO 9742566	A1	19971113	WO 97US8013	A	19970508	199751	B
AU 9730041	A	19971126	AU 9730041	A	19970508	199813	
US 5794025	A	19980811	US 96647375	A	19960509	199839	

Priority Applications (No Type Date): US 96647375 A 19960509
Cited Patents: US 4589087; US 4926374; US 5008807; US 5247627
Patent Details:
Patent No Kind Lan Pg Main IPC Filing Notes
WO 9742566 A1 E 77 G06F-009/302
Designated States (National): AL AM AU BA BB BG BR CA CN CU CZ EE FI GE HU IL IS JP KG KP KR LC LK LR LT LU LV MD MG MK MN MX NO NZ PL RO SG SI SK TR TT UA UG UZ VN
Designated States (Regional): AT BE CH DE DK EA ES FI FR GB GH GR IE IT KE LS LU MC MW NL OA PT SD SE SZ UG
AU 9730041 A G06F-009/302 Based on patent WO 9742566
US 5794025 A G06F-009/302

Abstract (Basic): WO 9742566 A

The method for processing cells in an asynchronous transfer mode (ATM) communication system involves using an ATM cell processor (12) which provides modulo arithmetic features which permits branching on the modulo portion of the result of an arithmetic operation. An arithmetic logic unit (ALU) or other processor instruction is modified to include a modulo field which specifies the number of right to left bits after which the result of the corresponding ALU operation is truncated.

Conditional **branch instructions** e.g. branch on zero result, branch on non-zero result, branch on carry and branch on overflow may be configured to operate only on the modulo portion of the ALU instruction result and/or on a carry out of the most significant bit (MSB) position of the modulo portion.

USE - **Performing** arithmetic operations in asynchronous transfer mode cell processing system.

ADVANTAGE - Provides improved cell scheduling and servicing techniques and improved ATM cell processor architecture.

Dwg.2/9

Title Terms: ASYNCHRONOUS; TRANSFER; MODE; CELL; PROCESS; METHOD; **EXECUTE** ; SUBSEQUENT; BRANCH; INSTRUCTION; BASED; MODULO; PORTION; RESULT; FIRST; INSTRUCTION; FIRST; INSTRUCTION; ARITHMETIC; OPERATE; INDICATE
Derwent Class: T01; W01
International Patent Class (Main): G06F-009/302
International Patent Class (Additional): G06F-009/32; H04J-003/24
File Segment: EPI

17/5/9 (Item 6 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

011025633 **Image available**
WPI Acc No: 1997-003557/199701
XRPX Acc No: N97-003140

Command appts for information processing appts - in which main command **sequence following branch instruction is used as preaddress for fetching following instruction when all contents of extracted static branch prediction information are not branched**

Patent Assignee: FUJITSU LTD (FUIT)
Number of Countries: 001 Number of Patents: 001
Patent Family:
Patent No Kind Date Applicat No Kind Date Week
JP 8272610 A 19961018 JP 9571534 A 19950329 199701 B

Priority Applications (No Type Date): JP 9571534 A 19950329
Patent Details:
Patent No Kind Lan Pg Main IPC Filing Notes
JP 8272610 A 14 G06F-009/38

Abstract (Basic): JP 8272610 A

The command appts has a program memory which stores the program consisting of a number of **command** sequence **instructions** . A **prefetch** control part (13) **performs** fetching of predetermined

command **execution** , from a main memory (21). The command sequence is stored in a **command** memory (12). A **prefetch** of **instruction** by a **prefetch** control part is **executed** from a branched detecting part (14). The presence of branch instruction in **command** sequence which is **prefetched** is detected. When a branch instruction is detected by branched detecting part, then a static branch prediction information corresponding to the branch instruction is extracted from a precommand sequence.

When all the contents of extracted static branch prediction information are not branched then the **main command** sequence which follows the branch instruction is used as preaddress for fetching the following **instruction** by **prefetch** control part.

ADVANTAGE - **Performs** static branch prediction effectively. Reduces increase in any hardware resources.

Dwg.1/3

Title Terms: COMMAND; APPARATUS; INFORMATION; PROCESS; APPARATUS; MAIN; COMMAND; SEQUENCE; FOLLOW; BRANCH; INSTRUCTION; FETCH; FOLLOW; INSTRUCTION; CONTENT; EXTRACT; STATIC; BRANCH; PREDICT; INFORMATION; BRANCH

Derwent Class: T01

International Patent Class (Main): G06F-009/38

File Segment: EPI

17/5/10 (Item 7 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2004 Thomson Derwent. All rts. reserv.

010451620 **Image available**

WPI Acc No: 1995-352938/199546

XRPX Acc No: N95-263148

Processing system operating method - processing instruction in response multiple branch instructions **prior to** execution **of** branch instructions , **and cancelling processing prior to completion in response to** execution **of any** branch instruction

Patent Assignee: INT BUSINESS MACHINES CORP (IBMC); IBM CORP (IBMC)

Inventor: SONG S P

Number of Countries: 005 Number of Patents: 006

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
EP 677807	A2	19951018	EP 95480021	A	19950321	199546	B
JP 7281894	A	19951027	JP 94317961	A	19941221	199601	
EP 677807	A3	19961023	EP 95480021	A	19950321	199648	
US 5644779	A	19970701	US 94228249	A	19940415	199732	
EP 677807	B1	19990303	EP 95480021	A	19950321	199913	
DE 69507975	E	19990408	DE 607975	A	19950321	199920	
			EP 95480021	A	19950321		

Priority Applications (No Type Date): US 94228249 A 19940415

Cited Patents: No-SR.Pub; 2.Jnl.Ref; JP 5040627; US 5461722

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
EP 677807	A2	E	21	G06F-009/38	
Designated States (Regional): DE FR GB					
JP 7281894	A		19	G06F-009/38	
US 5644779	A		19	G06F-009/38	
EP 677807	B1	E		G06F-009/38	
Designated States (Regional): DE FR GB					
DE 69507975	E			G06F-009/38	Based on patent EP 677807
EP 677807	A3			G06F-009/38	

Abstract (Basic): EP 677807 A

The method of operating a processing system involves processing an instruction prior to the **execution** of **branch instructions** in response to a number of **branch instructions** , and cancelling the processing of the instruction prior to completion of the **executed branch instruction** , in response to **execution** of any of the **branch**

instructions .

The cancelling step includes cancelling at least one basic block including the instruction and any **later instruction** processed prior to **execution** of the **executed branch instruction** . Each block is formed to include any instruction processed in response to a respective **branch instruction** and prior to a next **branch instruction** .

USE/ADVANTAGE - In superscalar processing system. Reduces delay in processing correct instruction after determining that branch is wrong.

Dwg.1/11

Title Terms: PROCESS; SYSTEM; OPERATE; METHOD; PROCESS; INSTRUCTION; RESPOND; MULTIPLE; BRANCH; INSTRUCTION; PRIOR; **EXECUTE** ; BRANCH; INSTRUCTION; CANCEL; PROCESS; PRIOR; COMPLETE; RESPOND; **EXECUTE** ; BRANCH ; INSTRUCTION

Derwent Class: T01

International Patent Class (Main): G06F-009/38

File Segment: EPI

17/5/11 (Item 8 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2004 Thomson Derwent. All rts. reserv.

010313752 **Image available**

WPI Acc No: 1995-215010/199528

XRPX Acc No: N95-168635

Storing interconnected branches of prefetch instructions **in cache - identifying** prefetched **Conditional Branch** Instructions , prefetching instructions **from one path and storing location of** first instruction **of other path, and nullifying stored addresses that are no longer in** execution **path**

Patent Assignee: NORTHERN TELECOM LTD (NELE)

Inventor: JAGER W J

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 5423048	A	19950606	US 92935941	A	19920827	199528	B

Priority Applications (No Type Date): US 92935941 A 19920827

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 5423048	A		10	G06F-013/00	

Abstract (Basic): US 5423048 A

An instruction **execution** tree comprising instructions is traversed in a predetermined manner. Instructions depending from both paths of a conditional branch **instruction** are **prefetched** .

When it is determined that a branch of **prefetched instructions** is not in the path of **execution** , the instructions associated with that branch are deleted, to prune the branch. Instruction addresses are selectively removed from a storage memory to provide the cache with instructions which will likely be required by the processor.

ADVANTAGE - Prunes unwanted branches of stored instructions, once it has been determined that they are not required by processor.

Dwg.3/3

Title Terms: STORAGE; INTERCONNECT; BRANCH; INSTRUCTION; CACHE; IDENTIFY; CONDITION; BRANCH; INSTRUCTION; INSTRUCTION; ONE; PATH; STORAGE; LOCATE; FIRST; INSTRUCTION; PATH; STORAGE; ADDRESS; NO; LONG; **EXECUTE** ; PATH

Derwent Class: T01

International Patent Class (Main): G06F-013/00

File Segment: EPI

17/5/12 (Item 9 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2004 Thomson Derwent. All rts. reserv.

010213889 **Image available**

WPI Acc No: 1995-115143/199515
XRPX Acc No: N95-090907
Servicing data cache miss in parallel with normal instruction execution **to avoid penalty - using existing load to general read only register** instruction **for** prefetching **data from main memory for storage in data cache memory before data cache miss occurs**
Patent Assignee: HEWLETT-PACKARD CO (HEWP)
Inventor: DELANO E R; FORSYTH M A
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 5396604	A	19950307	US 91729132	A	19910712	199515	B

Priority Applications (No Type Date): US 91729132 A 19910712
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 5396604	A		8	G06F-012/00	

Abstract (Basic): US 5396604 A
The computer-implemented method for servicing a data cache miss in parallel with normal instruction **execution** to thereby reduce or avoid the penalty associated with a data cache miss, involves
placing a **first instruction** that requires a load to a read only register within an instruction stream, wherein the **first instruction** provides a hint to the data cache memory to prefetch a cache line from a main memory into the data cache memory, and
decoding and **executing** the **first instruction** , wherein the load to the read only register does not alter the contents of the read only register.

Simultaneously with the prefetch of the cache line, further instructions that follow the **first instruction** are processed, such that the **first instruction** is placed within the instruction stream such that the cache line is prefetched from the main memory early enough so that subsequent accesses to this data will not result in a data cache miss.
USE/ADVANTAGE - Prefetching data into data cache by issuing existing load to read only register instruction. Permits data cache miss ''to be serviced'' in parallel with normal instruction **execution** .
Dwg.1/2
Title Terms: SERVICE; DATA; CACHE; MISS; PARALLEL; NORMAL; INSTRUCTION; **EXECUTE** ; AVOID; PENALTY; EXIST; LOAD; GENERAL; READ; REGISTER; INSTRUCTION; DATA; MAIN; MEMORY; STORAGE; DATA; CACHE; MEMORY; DATA; CACHE; MISS; OCCUR
Derwent Class: T01
International Patent Class (Main): G06F-012/00
File Segment: EPI

17/5/13 (Item 10 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

008882267 **Image available**
WPI Acc No: 1992-009536/199202
XRPX Acc No: N92-007331
Branching in pipelined processor - decodes instructions before executing instructions **to ensure optimal** performance **of code**
Patent Assignee: DIGITAL EQUIP CORP (DIGI); COMPAQ COMPUTER CORP (COPQ)
Inventor: SITES R L; WITEK R T
Number of Countries: 008 Number of Patents: 008
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
EP 463977	A	19920102	EP 91401783	A	19910628	199202	B
CA 2045791	A	19911230				199213	
TW 198109	A	19930111	TW 91106670	A	19910822	199325	

EP 463977	A3	19930922	EP 91401783	A	19910628	199509
EP 463977	B1	19980729	EP 91401783	A	19910628	199834
DE 69129881	E	19980903	DE 629881	A	19910628	199841
			EP 91401783	A	19910628	
KR 190252	B1	19990601	KR 9110880	A	19910628	200056
US 6167509	A	20001226	US 90547629	A	19900629	200103
			US 94243559	A	19940516	

Priority Applications (No Type Date): US 90547629 A 19900629; US 94243559 A 19940516
Cited Patents: NoSR.Pub; 3.Jnl.Ref; EP 219203
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
EP 463977	A				
Designated States (Regional): DE FR GB NL					
TW 198109	A			G06F-009/00	
EP 463977	B1	E		G06F-009/38	
Designated States (Regional): DE FR GB NL					
DE 69129881	E			G06F-009/38	Based on patent EP 463977
KR 190252	B1			G06F-009/38	
US 6167509	A			G06F-009/00	Cont of application US 90547629

Abstract (Basic): EP 463977 A

A pipelined processor is operated by fetching instructions from sequential addresses in **memory** and decoding the **instructions before executing** the **instructions** . The jump instructions are detected in the fetched instructions and the address of a register containing the address of the target of the jump instruction is extracted.

A predicted target address is also extracted from the register. An **instruction** is **prefetched** from the predicted target address rather than from the sequential addresses before the jump instruction is executed.

USE/ADVANTAGE - High performance processors executing a reduced instruction set. The processor employs a variable **memory** page size so that the entries in a translation **buffer** for implementing virtual addressing can be optimally used. (34pp Dwg.No.1/11)

Title Terms: BRANCH; PIPE; PROCESSOR; DECODE; INSTRUCTION; EXECUTE; INSTRUCTION; ENSURE; OPTIMUM; PERFORMANCE; CODE
Derwent Class: T01
International Patent Class (Main): G06F-009/00; G06F-009/38
International Patent Class (Additional): G06F-009/38
File Segment: EPI

17/5/14 (Item 11 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

008338536 **Image available**
WPI Acc No: 1990-225537/199030
XRPX Acc No: N90-175056
Production line method and appts. for instruction execution **- uses pipeline of instruction units that each** execute **and remove specific types of instruction**
Patent Assignee: BULL HN INFORMATION SYSTEMS INC (HONE)
Inventor: JOYCE T F; MIU M; MIU M T
Number of Countries: 021 Number of Patents: 016
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
EP 378816	A	19900725	EP 89123183	A	19891214	199030	B
AU 8946704	A	19900621				199031	
NO 8905068	A	19900620				199034	
CA 2005907	A	19900619				199036	
BR 8906579	A	19900904				199040	
FI 8906044	A	19900620				199040	
DK 8906481	A	19900620				199042	
ZA 8909554	A	19901031				199044	

CN 1044354	A	19900801				199117
IL 92605	A	19930131	IL 92605	A	19891208	199311
EP 378816	A3	19920506	EP 89123183	A	19891214	199330
CA 2005907	C	19940531	CA 2005907	A	19891218	199427
KR 9400027	B1	19940105	KR 8918859	A	19891218	199445
CN 1026037	C	19940928	CN 89109392	A	19891219	199539
EP 378816	B1	19970903	EP 89123183	A	19891214	199740
DE 68928300	E	19971009	DE 628300	A	19891214	199746
			EP 89123183	A	19891214	

Priority Applications (No Type Date): US 88286580 A 19881219
Cited Patents: NoSR.Pub; EP 198214; EP 236745; US 4598365
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
EP 378816	A				
Designated States (Regional): BE CH DE ES FR GB GR IT LI NL SE					
EP 378816	B1	E	30	G06F-009/38	
Designated States (Regional): DE FR GB IT					
DE 68928300	E			G06F-009/38	Based on patent EP 378816
IL 92605	A			G06F-009/38	
CA 2005907	C			G06F-009/38	
KR 9400027	B1			G06F-015/21	
CN 1026037	C			G06F-009/38	

Abstract (Basic): EP 378816 A

The data processing system consists of a CPU (2), a Virtual Memory **Management** Unit or VMMU (4), a Cache (6) and main memory (8). The CPU has three units which **execute** instructions in a production or pipe line fashion where each has a type of instruction it **executes** and removes from the production line.

The **first , instruction ,** unit (2-2) **prefetches instructions** from the I-cache, cracks them and **executes** any jump instructions and removes them from the production line. The second (2-4) **executes** instructions not requiring operands from main memory and removes them. It also sends the virtual address of any required operand to the VMMU. The last unit (2-6) receives instructions from the second and operands from the E-cache (6-4).

ADVANTAGE - Improved system throughput. (23pp Dwg.No.2/7

Title Terms: PRODUCE; LINE; METHOD; APPARATUS; INSTRUCTION; **EXECUTE** ; PIPE ; INSTRUCTION; UNIT; **EXECUTE** ; REMOVE; SPECIFIC; TYPE; INSTRUCTION
Derwent Class: T01
International Patent Class (Main): **G06F-009/38 ; G06F-015/21**
International Patent Class (Additional): **G06F-009/312 ; G06F-009/34 ; G06F-015/16**
File Segment: EPI

17/5/15 (Item 12 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

007275538
WPI Acc No: 1987-272545/198739
XRPX Acc No: N87-204129

Pipelined data processor for parallel processing - has pair of instruction registers storing two instructions to be executed

Patent Assignee: HITACHI LTD (HITA)
Inventor: INOUE K; KAMADA E; KURIYAMA K; SHINTANI Y; SHONAI T; YAMAOKA A
Number of Countries: 004 Number of Patents: 004
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
EP 239081	A	19870930	EP 87104345	A	19870324	198739	B
US 4858105	A	19890815	US 8730434	A	19870326	198941	
EP 239081	B1	19950906	EP 87104345	A	19870324	199540	
DE 3751503	G	19951012	DE 3751503	A	19870324	199546	
			EP 87104345	A	19870324		

Priority Applications (No Type Date): JP 8665651 A 19860326

Cited Patents: 5.Jnl.Ref; A3...9046; EP 21399; JP 57137944; No-SR.Pub
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
EP 239081	A	E	21		
Designated States (Regional): DE FR GB					
US 4858105	A		19		
EP 239081	B1	E	21	G06F-009/38	
Designated States (Regional): DE FR GB					
DE 3751503	G			G06F-009/38	Based on patent EP 239081

Abstract (Basic): EP 239081 A

An instruction extraction circuit (3) stores into the two instruction registers (1,2) the **first instruction** to be **executed** next and the second **following instruction** to be **executed** , from the main memory (13). A decoding control circuit detects whether either of the first and second instructions stored into the instruction registers would cause a conflict between the resources.

If the **first instruction** requires fetching of a memory operand and the use of the operation unit (12) and the second instruction requires the fetching of the instruction. There is no register conflict between those instructions so they can be **executed** in parallel because the operand buffer (10) and instruction buffer (11) are provided separately.

ADVANTAGE - Instruction **execution** time shortened by one cycle.

Title Terms: PIPE; DATA; PROCESSOR; PARALLEL; PROCESS; PAIR; INSTRUCTION; REGISTER; STORAGE; TWO; INSTRUCTION; **EXECUTE**
Derwent Class: T01
International Patent Class (Additional): **G06F-009/38 ; G06F-015/00**
File Segment: EPI

17/5/16 (Item 13 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

007261107
WPI Acc No: 1987-258114/198737
XRPX Acc No: N87-193271

Branch stream coprocessor for computer-executable instruction stream - has instructions pre - executed **in coprocessor which precedes pipeline processor and prepares instruction stream for input to processor**

Patent Assignee: IBM CORP (IBMC); INT BUSINESS MACHINES CORP (IBMC)
Inventor: BOMERENE J H; EMMA P; PUZAK T R; RECHTSCHAF R N; SPARCIO F J; EMMA P G; POMERENE J H; RECHTSCHAFFEN R N; SPARACIO F J
Number of Countries: 006 Number of Patents: 005
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
EP 236745	A	19870916	EP 87101590	A	19870205	198737	B
CA 1268555	A	19900501				199024	
US 4991080	A	19910205	US 86839312	A	19860313	199108	
EP 236745	B1	19950823	EP 87101590	A	19870205	199538	
DE 3751474	G	19950928	DE 3751474	A	19870205	199544	
			EP 87101590	A	19870205		

Priority Applications (No Type Date): US 86839312 A 19860313
Cited Patents: 3.Jnl.Ref; A3...9149; EP 106667; EP 229619; No-SR.Pub
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
EP 236745	A	E	30		
Designated States (Regional): DE FR GB IT					
EP 236745	B1	E	35	G06F-009/38	
Designated States (Regional): DE FR GB IT					
DE 3751474	G			G06F-009/38	Based on patent EP 236745

Abstract (Basic): EP 236745 A

The computing appts. for executing instructions in instruction sequences, the instruction sequences including first and second types

of instructions, the first type of instructions determining choices among alternative ones of the instruction sequence. The computing appts. comprises a **memory** for storing instructions and data. A main processor executes the instructions and processes the data. The **instructions** are **prefetched** from the **memory** and instruction sequences are formed for processing by the main processor.

A coprocessor, positioned between the prefetcher and the main processor and comprises a circuit for receiving the **instruction** sequences from the **prefetcher** and pre-executes ones of the first type of instructions.

ADVANTAGE - Prevents delays created by sharing of registers by consecutive instructions in instruction sequence.

3/17

Title Terms: BRANCH; STREAM; COMPUTER; EXECUTE; INSTRUCTION; STREAM; INSTRUCTION; PRE; EXECUTE; PRECEDE; PIPE; PROCESSOR; PREPARATION; INSTRUCTION; STREAM; INPUT; PROCESSOR

Derwent Class: T01

International Patent Class (Additional): G06F-009/38

File Segment: EPI

```
Set     Items   Description
S1          0   (COFETCH??? OR (CO OR PRE)()FETCH??? OR PREFETCH?)(3N)(COM-
             MAND? OR INSTRUCTION?)
S2          6   (PRIMARY OR PRIME OR FIRST OR 1ST OR INITIAL OR LEADING OR
             MAIN OR DOMINANT OR CARDINAL OR ORIGINAL)()(COMMAND? OR INSTR-
             UCTION?)
S3      32199   EXECUT? OR PERFORM? OR CONDUCT? OR MANAG? OR ACCOMPLISH?
S4          0   (PRE OR PRIOR OR BEFORE?)()(S3 (3N)(COMMAND? OR INSTRUCTIO-
             N?))
S5          7   (SECONDARY OR CHILD OR BRANCH OR LEAVES OR OFFSPRING OR OF-
             F()SPRING OR SUBPROGRAM OR SUBROUTINE OR SLAVE OR CALLED)(2W)-
             (COMMAND? OR INSTRUCTION?)
S6          2   (SEND? OR TRANSMIT? OR TRANSFER? OR CONVEY? OR DELIVER? OR
             OUTPUT? OR RETURN? OR DISPATCH?)(2W)((COMPUTER? OR CLIENT)(2N-
             )(NODE? OR PC OR PERSONAL()COMPUTER? OR WORKSTATION? OR WORK(-
             )STATION?))
S7          0   (POST OR AFTER OR SUBSEQUENT? OR LATER OR FOLLOWING)()(S3 -
             (3N) (COMMAND? OR INSTRUCTION))
S8          0   (STORING OR STORE? OR SAVE OR SAVING OR KEEP OR KEEPING OR
             PRESERV?)(3N) S5
S9       5549   CACHE OR BUFFER? OR STORAGE OR MEMORY OR REPOSITOR?
File 256:TecInfoSource 82-2004/Jul
         (c)2004 Info.Sources Inc
```

```
Set     Items   Description
S1        497   (COFETCH??? OR (CO OR PRE)()FETCH??? OR PREFETCH?)(3N)(COM-
                MAND? OR INSTRUCTION?)
S2        492   (PRIMARY OR PRIME OR FIRST OR 1ST OR INITIAL OR LEADING OR
                MAIN OR DOMINANT OR CARDINAL OR ORIGINAL)()(COMMAND? OR INSTR-
                UCTION?)
S3    5901867   EXECUT? OR PERFORM? OR CONDUCT? OR MANAG? OR ACCOMPLISH?
S4         74   (PRE OR PRIOR OR BEFORE?)()(S3 (3N)(COMMAND? OR INSTRUCTIO-
                N?))
S5       1395   (SECONDARY OR CHILD OR BRANCH OR LEAVES OR OFFSPRING OR OF-
                F()SPRING OR SUBPROGRAM OR SUBROUTINE OR SLAVE OR CALLED)(2W)-
                (COMMAND? OR INSTRUCTION?)
S6        736   (SEND? OR TRANSMIT? OR TRANSFER? OR CONVEY? OR DELIVER? OR
                OUTPUT? OR RETURN? OR DISPATCH?)(2W)((COMPUTER? OR CLIENT)(2N-
                )(NODE? OR PC OR PERSONAL()COMPUTER? OR WORKSTATION? OR WORK(-
                )STATION?))
S7        101   (POST OR AFTER OR SUBSEQUENT? OR LATER OR FOLLOWING)()(S3 -
                (3N) (COMMAND? OR INSTRUCTION))
S8         16   (STORING OR STORE? OR SAVE OR SAVING OR KEEP OR KEEPING OR
                PRESERV?)(3N) S5
S9     950914   CACHE OR BUFFER? OR STORAGE OR MEMORY OR REPOSITOR?
S10         6   S1 AND S2 AND S3
S11         4   S1 AND S4 AND S9
S12         0   S7 AND S3 AND S2 AND S5
S13         2   S3 AND S4 AND S5
S14         0   S3 AND S4 AND S6
S15       813   S3 AND S5
S16         0   (S4 OR S15) AND S6
S17         5   S4 AND S7
S18        17   S10 OR S11 OR S13 OR S17
S19        15   S18 NOT PY>2000
S20        15   S19 NOT PD>20001218
S21        12   RD (unique items)
File   8:Ei Compendex(R) 1970-2004/Jul W4
         (c) 2004 Elsevier Eng.  Info. Inc.
File  35:Dissertation Abs Online 1861-2004/May
         (c) 2004 ProQuest Info&Learning
File 202:Info. Sci. & Tech. Abs. 1966-2004/Jul 12
         (c) 2004 EBSCO Publishing
File  65:Inside Conferences 1993-2004/Aug W1
         (c) 2004 BLDSC all rts. reserv.
File   2:INSPEC 1969-2004/Jul W4
         (c) 2004 Institution of Electrical Engineers
File 233:Internet & Personal Comp. Abs. 1981-2003/Sep
         (c) 2003 EBSCO Pub.
File  94:JICST-EPlus 1985-2004/Jul W2
         (c)2004 Japan Science and Tech Corp(JST)
File  99:Wilson Appl. Sci & Tech Abs 1983-2004/Jul
         (c) 2004 The HW Wilson Co.
File  95:TEME-Technology & Management 1989-2004/Jun W1
         (c) 2004 FIZ TECHNIK
File 583:Gale Group Globalbase(TM) 1986-2002/Dec 13
         (c) 2002 The Gale Group
```

21/5/1 (Item 1 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)
(c) 2004 Elsevier Eng. Info. Inc. All rts. reserv.

05803555 E.I. No: EIP00025064770
Title: Register allocation by systematic merge of register-reuse chains
Author: Zhang, Yukong; Lee, Hyuk Jae
Corporate Source: Louisiana Tech Univ, Ruston, LA, USA
Source: Electronics Letters v 36 n 1 Jan 2000. p 16-17
Publication Year: 2000
CODEN: ELLEAK ISSN: 0013-5194
Language: English
Document Type: JA; (Journal Article) Treatment: T; (Theoretical)
Journal Announcement: 0104W3
Abstract: If register allocation is **performed after instruction** scheduling, the efficiency of register allocation can be degraded due to constraints caused by such scheduling. The authors propose a new register allocation technique that is **performed before instruction** scheduling and, consequently, allows great freedom in the optimization of register allocation. Experiments show that the efficiency of the proposed technique is on average 7.4% greater than that of the conventional approach. (Author abstract) 5 Refs.
Descriptors: *Distributed computer systems; Resource allocation; Optimization
Identifiers: Register allocation
Classification Codes:
722.4 (Digital Computers & Systems); 912.2 (Management); 921.5 (Optimization Techniques)
722 (Computer Hardware); 912 (Industrial Engineering & Management); 921 (Applied Mathematics)
72 (COMPUTERS & DATA PROCESSING); 91 (ENGINEERING MANAGEMENT); 92 (ENGINEERING MATHEMATICS)

21/5/2 (Item 2 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)
(c) 2004 Elsevier Eng. Info. Inc. All rts. reserv.

05192425 E.I. No: EIP98124511039
Title: Cooperative prefetching: Compiler and hardware support for effective instruction prefetching **in modern processors**
Author: Luk, Chi-Keung; Mowry, Todd C.
Corporate Source: Univ of Toronto, Toronto, Ont, Can
Conference Title: Proceedings of the 1998 31st Annual ACM/IEEE International Symposium on Microarchitecture
Conference Location: Dallas, TX, USA Conference Date: 19981130-19981202
Sponsor: IEEE
E.I. Conference No.: 49389
Source: Proceedings of the Annual International Symposium on Microarchitecture 1998. IEEE Comp Soc, Los Alamitos, CA, USA. p 182-193
Publication Year: 1998
CODEN: PSMIE7 ISSN: 1072-4451
Language: English
Document Type: CA; (Conference Article) Treatment: A; (Applications); T ; (Theoretical)
Journal Announcement: 9902W3
Abstract: Instruction cache miss latency is becoming an increasingly important **performance** bottleneck, especially for commercial applications. Although **instruction prefetching** is an attractive technique for tolerating this latency, we find that existing prefetching schemes are insufficient for modern superscalar processors since they fail to issue prefetches early enough (particularly for non-sequential accesses). To overcome these limitations, we propose a new **instruction prefetching** technique whereby the hardware and software cooperate to hide the latency as follows. The hardware **performs** aggressive sequential prefetching combined with a novel prefetches filtering mechanism to allow it to get far ahead without polluting the cache. To hide the latency of non-sequential accesses, we propose and implement a novel compiler algorithm which

automatically inserts **instruction - prefetch instructions** into the **executable** to prefetch the targets of control transfers far enough in advance. Our experimental results demonstrate that this new approach results in speedups ranging from 9.4% to 18.5% (13.3% on average) over the original **execution** time on an out-of-order superscalar processor, which is more than double the average speedup of the best existing schemes (6.5%). This is **accomplished** by hiding an average of 71% of the **original instruction** stall time, compared with only 36% for the best existing schemes. We find that both the prefetch filtering and compiler-inserted prefetching components of our design are essential and complementary, that the compiler can limit the code expansion to less than 10% on average, and that our scheme is robust with respect to variations in miss latency and bandwidth. (Author abstract) 15 Refs.

Descriptors: *Reduced instruction set computing; Microprogramming; Microprocessor chips; Program compilers; Computer hardware; Response time (computer systems); Signal filtering and prediction; Storage allocation (computer); Buffer storage; Bandwidth

Identifiers: Superscalar processors; Cooperative prefetching

Classification Codes:

723.1 (Computer Programming); 714.2 (Semiconductor Devices & Integrated Circuits); 722.4 (Digital Computers & Systems); 716.1 (Information & Communication Theory); 722.1 (Data Storage, Equipment & Techniques)

722 (Computer Hardware); 723 (Computer Software); 714 (Electronic Components); 716 (Radar, Radio & TV Electronic Equipment)

72 (COMPUTERS & DATA PROCESSING); 71 (ELECTRONICS & COMMUNICATIONS)

21/5/3 (Item 3 from file: 8)

DIALOG(R)File 8:Ei Compendex(R)
(c) 2004 Elsevier Eng. Info. Inc. All rts. reserv.

04862229 E.I. No: EIP97103904586

Title: Improving data cache performance **by** pre - executing instructions **under a** cache **miss**

Author: Dundas, James; Mudge, Trevor

Corporate Source: Univ of Michigan, Ann Arbor, MI, USA

Conference Title: Proceedings of the 1997 International Conference on Supercomputing

Conference Location: Vienna, Austria Conference Date: 19970707-19970711

Sponsor: ACM

E.I. Conference No.: 47178

Source: Proceedings of the International Conference on Supercomputing 1997. ACM, New York, NY, USA. p 68-75

Publication Year: 1997

CODEN: 002151

Language: English

Document Type: CA; (Conference Article) Treatment: A; (Applications)

Journal Announcement: 9712W4

Abstract: In this paper we propose and evaluate a technique that improves first level data **cache** performance by **pre - executing** future **instructions** under a data **cache** miss. We show that these **pre - executed instructions** can generate highly accurate data prefetches, particularly when the first level **cache** is small. The technique is referred to as runahead processing. The hardware required to implement runahead is modest, because, when a miss occurs, it makes use of an otherwise idle resource, the execution logic. The principal hardware cost is an extra register file. To measure the impact of runahead, we simulated a processor executing five integer Spec95 benchmarks. Our results show that runahead was able to significantly reduce data **cache** CPI for four of the five benchmarks. We also compared runahead to a simple form of prefetching, sequential prefetching, which would seem to be suitable for scientific benchmarks. We confirm this by enlarging the scope of our experiments to include a scientific benchmark. However, we show that runahead was also able to outperform sequential prefetching on the scientific benchmark. We also conduct studies that demonstrate that runahead can generate many useful prefetches for lines that show little spatial locality with the misses that initiate runahead episodes. Finally, we discuss some further enhancements of our baseline runahead prefetching scheme. (Author abstract) 15 Refs.

Descriptors: Data **storage** equipment; Program compilers; Computer software; **Storage** allocation (computer); Computer simulation
Identifiers: Runahead processing; **Prefetching** ; **Instruction stride table**
Classification Codes:
722.1 (Data Storage, Equipment & Techniques); 723.1 (Computer Programming); 723.5 (Computer Applications)
722 (Computer Hardware); 723 (Computer Software)
72 (COMPUTERS & DATA PROCESSING)

21/5/4 (Item 4 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)
(c) 2004 Elsevier Eng. Info. Inc. All rts. reserv.

04426164 E.I. No: EIP96063214925
Title: Instruction prefetching **of systems codes with layout optimized for reduced cache misses**
Author: Xia, Chun; Torrellas, Josep
Corporate Source: Sun Microsystems, Inc, Urbana, IL, USA
Conference Title: Proceedings of the 1996 23rd Annual International Symposium on Computer Architecture
Conference Location: Philadelphia, PA, USA Conference Date: 19960522-19960524
Sponsor: ACM; IEEE
E.I. Conference No.: 44779
Source: Conference Proceedings - Annual International Symposium on Computer Architecture 1996. IEEE, Los Alamitos, CA, USA. p 271-282
Publication Year: 1996
CODEN: CPAADU ISSN: 0884-7495
Language: English
Document Type: CA; (Conference Article) Treatment: A; (Applications); T ; (Theoretical)
Journal Announcement: 9608W2
Abstract: High- **performing** on-chip instruction caches are crucial to keep fast processors busy. Unfortunately, while on-chip caches are usually successful at intercepting instruction fetches in loop-intensive engineering codes, they are less able to do so in large systems codes. To improve the **performance** of the latter codes, the compiler can be used to lay out the code in memory for reduced cache conflicts. Interestingly, such an operation leaves the code in a state that can be exploited by a new type of **instruction prefetching** : guarded sequential **prefetching** . The idea is that the compiler leaves hints in the code as to how the code was laid out. Then, at run time, the prefetching hardware detects these hints and uses them to prefetch more effectively. This scheme can be implemented very cheaply: one bit encoded in control transfer **instructions** and a **prefetch** module that requires minor extensions to existing next-line sequential prefetches. Furthermore, the scheme can be turned off and on at run time with the toggling of a bit in the TLB. The scheme is evaluated with simulations using complete traces from a 4-processor machine. Overall, for 16-Kbyte **primary instruction** caches, guarded sequential **prefetching** removes, on average, 66% of the instruction misses remaining in an operating system with an optimized layout, speeding up the operating system by 10%. Moreover, the scheme is more cost-effective and robust than existing sequential prefetching techniques. (Author abstract) 14 Refs.
Descriptors: *Program compilers; Data storage equipment; Computer hardware; Program processors; Computer operating systems; Computer simulation; Multiprocessing systems; UNIX; Algorithms; Optimization
Identifiers: Sequential **prefetching** techniques; **Instruction** memory hierarchy; **Instruction prefetching**
Classification Codes:
723.1 (Computer Programming); 722.1 (Data Storage, Equipment & Techniques); 722.2 (Computer Peripheral Equipment); 723.5 (Computer Applications); 722.4 (Digital Computers & Systems); 921.5 (Optimization Techniques)
723 (Computer Software); 722 (Computer Hardware); 921 (Applied Mathematics)
72 (COMPUTERS & DATA PROCESSING); 92 (ENGINEERING MATHEMATICS)

21/5/5 (Item 1 from file: 35)
DIALOG(R)File 35:Dissertation Abs Online
(c) 2004 ProQuest Info&Learning. All rts. reserv.

1022175 ORDER NO: AAD88-17099
AN APPROACH TO PHASE-COUPLED RESOURCE ALLOCATION AND CODE REORGANIZATION FOR A CLASS OF PIPELINE ARCHITECTURES
Author: ARDOIN, CY DALE
Degree: PH.D.
Year: 1988
Corporate Source/Institution: UNIVERSITY OF SOUTHWESTERN LOUISIANA (0233)
DIRECTOR: SUBRATA DASGUPTA
Source: VOLUME 49/07-B OF DISSERTATION ABSTRACTS INTERNATIONAL.
PAGE 2716. 110 PAGES
Descriptors: COMPUTER SCIENCE
Descriptor Codes: 0984

The advent of instruction pipelines without interlock pipe stages (MIPS) and those with delayed branches and potential resource conflicts has, in recent years, created the need for code reorganizers. One critical issue in the design of code reorganizers is the identification and exploitation of parallelism.

In this dissertation, we examine two of the phases involved in code translation. These phases are resource management and instruction scheduling. The most notable problem between these two phases is their mutual dependence. Namely, the resource binding made by the resource manager is dependent on the order in which the instructions execute. But clearly, the order in which the instructions should execute is dependent on the resources used by the instructions.

Because of this mutual dependence a coupled approach to resource management and instruction scheduling is indicated. Through this coupling process the scheduling phase will have a greater opportunity to produce quality code. This is primarily a result of the elimination of resource dependencies introduced during the register allocation phase. These resource dependencies impair the instruction scheduler's ability to recognize parallelism in the program.

To fully assess the virtues of a phase-coupled system, it is necessary to compare and contrast different approaches to the problem of code reorganization. The four variations to be examined differ in the order in which they perform the resource management and instructions scheduling phases. The four variations are: resource **management before instruction** scheduling (traditional), resource **management before instruction** scheduling (dynamic dependency arcs), resource **management after instruction** scheduling, and resource **management** phase-coupled with **instruction** scheduling.

21/5/6 (Item 2 from file: 35)
DIALOG(R)File 35:Dissertation Abs Online
(c) 2004 ProQuest Info&Learning. All rts. reserv.

931699 ORDER NO: AAD86-24207
THE USE OF BEST EXAMPLE THEORY IN THE ACQUISITION OF SOCIAL SKILLS BY ADULTS WITH MENTAL RETARDATION: A SYSTEMATIC REPLICATION
Author: BECKWITH, RUTHIE-MARIE
Degree: PH.D.
Year: 1986
Corporate Source/Institution: GEORGE PEABODY COLLEGE FOR TEACHERS OF VANDERBILT UNIVERSITY (0074)
Source: VOLUME 47/07-A OF DISSERTATION ABSTRACTS INTERNATIONAL.
PAGE 2539. 113 PAGES
Descriptors: EDUCATION, SPECIAL
Descriptor Codes: 0529

The purpose of this study was to investigate the use of best example

theory in the instruction of social skills with mentally retarded adults. This study consisted of three parts: establishing the ability of normal persons to discriminate good, moderate, and poor videotaped exemplars of social skills; instruction of social skill discrimination; and measuring the impact of instruction on the performance of social skills.

Videotaped exemplars of six target behaviors were developed; greetings, exit amenity, interview, requesting assistance, responding to criticism, and responding to an unreasonable request. These exemplars were rated by volunteers using a 7-point Likert-type scale to determine their goodness-of-example (GOE). Mean rating scores for each exemplar were calculated to verify the existence of a range of exemplars for each target behavior.

Intervention with subjects began with a pretest for social skill discrimination. Instruction for three of the target behaviors using the exemplars identified during the rating process was conducted under three experimental conditions using a repeated measures Latin square design. These conditions consisted of the presentation of three good exemplars (GE3), one good exemplar (GE1), or one good, moderate, and poor exemplar (ALL). Posttests for acquisition of the trained target behaviors and generalization to the untrained target behaviors were then conducted. An analysis of variance for the repeated measures Latin square design was conducted for the training component of the study.

The impact of instruction on the subjects' performance of the target behavior interview was measured by videotaping the subjects during interviews **conducted before** and **after instruction** . The videotapes of the subjects were rated by judges using a 7-point Likert-type scale. Mean ratings for each performance were analyzed using a Spearman rank correlation coefficient (Siegel, 1956) to determine if a change in performance was evident.

Results of this study established the ability of normal persons to distinguish between social skills exemplars using a best example approach. The use of only one good exemplar during instruction was sufficient for the acquisition of discrimination. The ability to discriminate did not generalize to the actual performance of a targeted behavior.

21/5/7 (Item 1 from file: 2)

DIALOG(R)File 2:INSPEC
(c) 2004 Institution of Electrical Engineers. All rts. reserv.

6491682 INSPEC Abstract Number: C2000-03-5220P-022

Title: Register allocation by systematic merge of register-reuse chains

Author(s): Yukong Zhang; Hyuk Jae Lee
Author Affiliation: Louisiana Tech. Univ., Ruston, LA, USA

Journal: Electronics Letters vol.36, no.1 p.16-17
Publisher: IEE,
Publication Date: 6 Jan. 2000 Country of Publication: UK
CODEN: ELLEAK ISSN: 0013-5194
SICI: 0013-5194(20000106)36:1L.16:RASM;1-4
Material Identity Number: E089-2000-001
U.S. Copyright Clearance Center Code: 0013-5194/2000/$10.00
Language: English Document Type: Journal Paper (JP)
Treatment: Practical (P); Experimental (X)

Abstract: If register allocation is **performed after instruction** scheduling, the efficiency of register allocation can be degraded due to constraints caused by such scheduling. The authors propose a new register allocation technique that is **performed before instruction** scheduling and, consequently, allows great freedom in the optimisation of register allocation. Experiments show that the efficiency of the proposed technique is on average 7.4%, greater than that of the conventional approach. (5 Refs)

Subfile: C
Descriptors: circuit optimisation; embedded systems; graph theory; parallel architectures
Identifiers: register allocation; systematic merge; register-reuse chains ; embedded processors; optimisation; efficiency
Class Codes: C5220P (Parallel architecture); C1160 (Combinatorial

mathematics); C1180 (Optimisation techniques)
Copyright 2000, IEE

21/5/8 (Item 2 from file: 2)
DIALOG(R)File 2:INSPEC
(c) 2004 Institution of Electrical Engineers. All rts. reserv.

6129249 INSPEC Abstract Number: C1999-02-5470-015
Title: Cooperative prefetching: compiler and hardware support for effective instruction prefetching **in modern processors**
Author(s): Chi-Keung Luk; Mowry, T.C.
Author Affiliation: Dept. of Comput. Sci., Toronto Univ., Ont., Canada
Conference Title: Proceedings. 31st Annual ACM/IEEE International Symposium on Microarchitecture p.182-93
Publisher: IEEE Comput. Soc, Los Alamitos, CA, USA
Publication Date: 1998 Country of Publication: USA xiv+317 pp.
ISBN: 0 8186 8609 X Material Identity Number: XX-1998-03444
U.S. Copyright Clearance Center Code: 0 8186 8609 X/98/$10.00
Conference Title: Proceedings. 31st Annual ACM/IEEE International Symposium on Microarchitecture
Conference Sponsor: IEEE Comput. Soc. Tech. Comm. on Microprogramming & Microarchit.; ACM SIGMICRO
Conference Date: 30 Nov.-2 Dec. 1998 Conference Location: Dallas, TX, USA
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P)
Abstract: Instruction cache miss latency is becoming an increasingly important **performance** bottleneck, especially for commercial applications. Although **instruction prefetching** is an attractive technique for tolerating this latency, we find that existing prefetching schemes are insufficient for modern superscalar processors since they fail to issue prefetches early enough (particularly for non-sequential accesses). To overcome these limitations, we propose a new **instruction prefetching** technique whereby the hardware and software cooperate to hide the latency as follows. The hardware **performs** aggressive sequential prefetching combined with a novel prefetch filtering mechanism to allow it to get far ahead without polluting the cache. To hide the latency of non-sequential accesses, we propose and implement a novel compiler algorithm which automatically inserts **instruction prefetch instructions** into the **executable** to prefetch the targets of control transfers far enough in advance. Our experimental results demonstrate that this new approach results in speedups ranging from 9.4% to 18.5% (13.3% on average) over the original **execution** time on an out-of-order superscalar processor; which is more than double the average speedup of the best existing schemes (6.5%). This is **accomplished** by hiding an average of 71% of the **original instruction** stall time, compared with only 36% for the best existing schemes. We find that both the prefetch filtering and compiler-inserted prefetching components of our design are essential and complementary, that the compiler can limit the code expansion to less than 10% on average, and that our scheme is robust with respect to variations in miss latency and bandwidth. (15 Refs)
Subfile: C
Descriptors: computer architecture; **performance** evaluation; program compilers
Identifiers: **instruction prefetching** ; cache miss latency; **performance** bottleneck; prefetch filtering; compiler-inserted prefetching; compiler
Class Codes: C5470 (Performance evaluation and testing); C5220 (Computer architecture); C6150C (Compilers, interpreters and other processors)
Copyright 1999, IEE

21/5/9 (Item 3 from file: 2)
DIALOG(R)File 2:INSPEC
(c) 2004 Institution of Electrical Engineers. All rts. reserv.

5304367 INSPEC Abstract Number: C9608-5220-004
Title: Instruction prefetching **of systems codes with layout optimized for reduced cache misses**
Author(s): Chun Xia; Torrellas, J.
Author Affiliation: Center for Supercomput. Res. & Dev., Illinois Univ., Urbana, IL, USA
Journal: Computer Architecture News Conference Title: Comput. Archit. News (USA) vol.24, no.2 p.271-82
Publisher: ACM,
Publication Date: May 1996 Country of Publication: USA
CODEN: CANED2 ISSN: 0163-5964
SICI: 0163-5964(199605)24:2L.271:IPSC;1-M
Material Identity Number: B580-96003
U.S. Copyright Clearance Center Code: 0 89791 786 3/96/0005.$3.50
Conference Title: ISCA '96: The 23rd Annual International Conference on Computer Architecture
Conference Sponsor: ACM; IEEE Comput. Soc.; TCCA
Conference Date: 22-24 May 1996 Conference Location: Philadelphia, PA, USA
Language: English Document Type: Conference Paper (PA); Journal Paper (JP)
Treatment: Practical (P)
Abstract: High- **performing** on-chip instruction caches are crucial to keep fast processors busy. Unfortunately while on-chip caches are usually successful at intercepting instruction fetches in loop-intensive engineering codes, they are less able to do so in large systems codes. To improve the **performance** of the latter codes, the compiler can be used to lay out the code in memory for reduced cache conflicts. Interestingly, such an operation leaves the code in a state that can be exploited by a new type of **instruction prefetching** : guarded sequential **prefetching** . The idea is that the compiler leaves hints in the code as to how the code was laid out. Then, at run time, the prefetching hardware detects these hints and uses them to prefetch more effectively. This scheme can be implemented very cheaply: one bit encoded in control transfer **instructions** and a **prefetch** module that requires minor extensions to existing next-line sequential prefetchers. Furthermore, the scheme can be turned off and on at run time with the toggling of a bit in the TLB. The scheme is evaluated with simulations using complete traces from a 4-processor machine. Overall, for 16-Kbyte **primary instruction** caches, guarded sequential **prefetching** removes, on average, 66% of the instruction misses remaining in an operating system with an optimized layout, speeding up the operating system by 10% Moreover, the scheme is more cost-effective and robust than existing sequential prefetching techniques. (14 Refs)
Subfile: C
Descriptors: computer architecture; instruction sets; processor scheduling; program compilers
Identifiers: **instruction prefetching** ; systems codes; reduced cache misses; intercepting instruction fetches; loop-intensive engineering codes; **performance** ; compiler; guarded sequential prefetching; 16-Kbyte **primary instruction** caches
Class Codes: C5220 (Computer architecture); C6150C (Compilers, interpreters and other processors); C6150N (Distributed systems software)
Copyright 1996, IEE

21/5/10 (Item 1 from file: 94)
DIALOG(R)File 94:JICST-EPlus
(c)2004 Japan Science and Tech Corp(JST). All rts. reserv.

02558893 JICST ACCESSION NUMBER: 95A0825654 FILE SEGMENT: JICST-E
Data prefetch scheme by Hardware unit executed LOAD pre-execution.
NAKASUMI MITSUAKI (1); HORIGUCHI SUSUMU (1); OKAMOTO SHUSUKE (1); SOWA MASAHIRO (1)
(1) Univ. of Electro-Communications, Grad. Sch.
Joho Shori Gakkai Kenkyu Hokoku, 1995, VOL.95,NO.80(ARC-113), PAGE.161-167, FIG.3, TBL.1, REF.5
JOURNAL NUMBER: Z0031BAO ISSN NO: 0919-6072
UNIVERSAL DECIMAL CLASSIFICATION: 681.326

LANGUAGE: Japanese COUNTRY OF PUBLICATION: Japan
DOCUMENT TYPE: Journal
ARTICLE TYPE: Original paper
MEDIA TYPE: Printed Publication
ABSTRACT: Distributed shared **memory** systems are remarkable, Because these Parallel systems accept exist programs with minimum modify. But Distributed shared **memory** systems have a disadvantage as data latency brought by both the size of these systems and the speed gap between **memory** and processor. To hide reference latency on a Distributed shared **memory** system, Software Prefetching has been widely used. but this method requires overhead to issue **prefetch instructions** . We have already proposed a new Hardware unit based on data prefetching scheme for these systems. The basic idea is to place the Hardware unit which is able to **execute** load **instructions before executed** by processor. In this paper, We argue that how the unit executes prefetching when it meets conditioned Branch. because it's important to make full use of its ability. and We make a simulation of the Hardware unit on a CAD called PARTHENON. (author abst.)
DESCRIPTORS: advanced control; parallel computer; system description language; computer simulation; speedup; logical design; synthesis; **storage** system; distributed processing; distributed **memory** ; shared **memory**
BROADER DESCRIPTORS: instruction control; control; control system(computer) ; method; digital computer; computer; hardware; programming language; formal language; language; computer application; utilization; simulation; modification; improvement; design; treatment
CLASSIFICATION CODE(S): JC02030K

21/5/11 (Item 2 from file: 94)
DIALOG(R)File 94:JICST-EPlus
(c)2004 Japan Science and Tech Corp(JST). All rts. reserv.

01328894 JICST ACCESSION NUMBER: 91A0721103 FILE SEGMENT: JICST-E
A Superscalar Architecture using Boosting and Delayed Branch.
ANDO HIDEKI (1); NAKANISHI CHIKAKO (1); NAKAYA MASAO (1)
(1) Mitsubishi Electric Corp.
Joho Shori Gakkai Kenkyu Hokoku, 1991, VOL.91,NO.64(ARC-89), PAGE.33-40, FIG.6, TBL.1, REF.12
JOURNAL NUMBER: Z0031BAO ISSN NO: 0919-6072
UNIVERSAL DECIMAL CLASSIFICATION: 681.32
LANGUAGE: Japanese COUNTRY OF PUBLICATION: Japan
DOCUMENT TYPE: Journal
ARTICLE TYPE: Original paper
MEDIA TYPE: Printed Publication
ABSTRACT: Exploiting instruction-level parallelism and alleviation of **performance** degradation by branch delay are issues which should be solved without cycle time penalty. We adopt instruction boosting M. Smith 90! to extract instruction-level parallelism with a simple hardware. And a delayed **branch** with an **instruction** queue reduces branch delay. Delay slots are filled by **branch** target **instructions** , and are **executed** when branch is taken. Sequential successors after delay slots are pre-fetched in the queue **before execution** of a **branch instruction** , and **executed** when branch is not taken. **Performance** evaluation shows that the speedup is 1.6-1.8 times over scalar machines. (author abst.)
DESCRIPTORS: computer architecture; parallel processing; instruction control; scheduling; pipeline processing; benchmark(computer); **performance** ; system evaluation; schematic design
BROADER DESCRIPTORS: computer system(architecture); method; treatment; control; control system(computer); evaluation; design
CLASSIFICATION CODE(S): JC02010O

21/5/12 (Item 3 from file: 94)
DIALOG(R)File 94:JICST-EPlus
(c)2004 Japan Science and Tech Corp(JST). All rts. reserv.

00691141 JICST ACCESSION NUMBER: 88A0536178 FILE SEGMENT: JICST-E

A branch method of parallel neumann computer.

KAWAMURA TADAAKI (1); SOWA MASAHIRO (1)
(1) Nagoya Inst. of Technology

Denshi Joho Tsushin Gakkai Gijutsu Kenkyu Hokoku(IEIC Technical Report (Institute of Electronics, Information and Communication Enginners), 1988, VOL.88,NO.155, PAGE.67-72(CPSY88-36), FIG.8, REF.8
JOURNAL NUMBER: S0532BBG
UNIVERSAL DECIMAL CLASSIFICATION: 681.32+
LANGUAGE: Japanese COUNTRY OF PUBLICATION: Japan
DOCUMENT TYPE: Journal
ARTICLE TYPE: Original paper
MEDIA TYPE: Printed Publication

ABSTRACT: We present a Branch method of a so called PN(Parallel Neumann) computer. The PN computer is a computer, proposed by us, which **executes** von Neumann programs in parallel. That is, it classifies the instructions of the von Neumann computer into three groups (data transference, arithmetic calculation, and branch), and **executes** them separately. The branch on PN computer has special features; prefetch and partial or complete **pre - execution** of the **branch instruction** .(author abst.)

DESCRIPTORS: instruction set; branching; parallel computer; parallel processing; register; assembly language; token; data transfer; arithmetic processor; operation(mathematics); queue; computer algorithm
BROADER DESCRIPTORS: digital computer; computer; hardware; treatment; programming language; formal language; language; queuing theory; theory ; algorithm
CLASSIFICATION CODE(S): JC02050G

```
Set     Items   Description
S1        738   (COFETCH??? OR (CO OR PRE)()FETCH??? OR PREFETCH?)(3N)(COM-
              MAND? OR INSTRUCTION?)
S2       2401   (PRIMARY OR PRIME OR FIRST OR 1ST OR INITIAL OR LEADING OR
              MAIN OR DOMINANT OR CARDINAL OR ORIGINAL)()(COMMAND? OR INSTR-
              UCTION?)
S3   13744927   EXECUT? OR PERFORM? OR CONDUCT? OR MANAG? OR ACCOMPLISH?
S4        210   (PRE OR PRIOR OR BEFORE?)()(S3 (3N)(COMMAND? OR INSTRUCTIO-
              N?))
S5       3382   (SECONDARY OR CHILD OR BRANCH OR LEAVES OR OFFSPRING OR OF-
              F()SPRING OR SUBPROGRAM OR SUBROUTINE OR SLAVE OR CALLED)(2W)-
              (COMMAND? OR INSTRUCTION?)
S6       2258   (SEND? OR TRANSMIT? OR TRANSFER? OR CONVEY? OR DELIVER? OR
              OUTPUT? OR RETURN? OR DISPATCH?)(2W)((COMPUTER? OR CLIENT)(2N-
              )(NODE? OR PC OR PERSONAL()COMPUTER? OR WORKSTATION? OR WORK(-
              )STATION?))
S7        616   (POST OR AFTER OR SUBSEQUENT? OR LATER OR FOLLOWING)()(S3 -
              (3N) (COMMAND? OR INSTRUCTION))
S8         71   (STORING OR STORE? OR SAVE OR SAVING OR KEEP OR KEEPING OR
              PRESERV?)(3N) S5
S9    1604577   CACHE OR BUFFER? OR STORAGE OR MEMORY OR REPOSITOR?
S10         6   S1 (S) S2 (S) S3
S11         3   S1 (S) S4 (S) S9
S12         0   S7 (S) S3 (S) S2 (S) S5
S13         7   S3 (S) S4 (S) S5
S14         0   S3 (S) S4 (S) S6
S15      1332   S3 (S) S5
S16         0   (S4 OR S15) AND S6
S17         5   S4 (S) S7
S18        21   S10 OR S11 OR S13 OR S17
S19        16   S18 NOT PY>2000
S20        16   S19 NOT PD>20001218
S21        15   RD (unique items)
File  15:ABI/Inform(R) 1971-2004/Aug 05
         (c) 2004 ProQuest Info&Learning
File 810:Business Wire 1986-1999/Feb 28
         (c) 1999 Business Wire
File 647:CMP  Computer Fulltext 1988-2004/Jul W4
         (c) 2004 CMP Media, LLC
File 275:Gale Group Computer DB(TM) 1983-2004/Aug 06
         (c) 2004 The Gale Group
File 674:Computer News Fulltext 1989-2004/Jul W4
         (c) 2004 IDG Communications
File 696:DIALOG Telecom. Newsletters 1995-2004/Aug 05
         (c) 2004 The Dialog Corp.
File 621:Gale Group New Prod.Annou.(R) 1985-2004/Aug 06
         (c) 2004 The Gale Group
File 636:Gale Group Newsletter DB(TM) 1987-2004/Aug 06
         (c) 2004 The Gale Group
File 813:PR Newswire 1987-1999/Apr 30
         (c) 1999 PR Newswire Association Inc
File 613:PR Newswire 1999-2004/Aug 05
         (c) 2004 PR Newswire Association Inc
File  16:Gale Group PROMT(R) 1990-2004/Aug 06
         (c) 2004 The Gale Group
File 160:Gale Group PROMT(R) 1972-1989
         (c) 1999 The Gale Group
File 553:Wilson Bus. Abs. FullText 1982-2004/Jul
         (c) 2004 The HW Wilson Co
```

21/3,K/1 (Item 1 from file: 15)
DIALOG(R)File 15:ABI/Inform(R)
(c) 2004 ProQuest Info&Learning. All rts. reserv.

02379335 84988409
MUIApp: an object-oriented graphical user interface application framework
Rajagopala, Mukunda G.; Hsieh, Shang-Hsien; Sotelino, Elisa D.; White, Donald W.
Engineering Computations v14n3 PP: 256-280 1997
ISSN: 0264-4401 JRNL CODE: NGCP
WORD COUNT: 6432

...TEXT: the command objects are complete and self-contained, they can be stored in lists, and queued for **later execution** .

By representing the **commands** as objects, it becomes easier to implement the "undo" action by simply implementing an additional function that...

... the "do" action of the command. Also, it may be necessary to save some state of the **command before executing** the "do" or "undo" action. This information can be stored in a data member of the class...

21/3,K/2 (Item 2 from file: 15)
DIALOG(R)File 15:ABI/Inform(R)
(c) 2004 ProQuest Info&Learning. All rts. reserv.

01356458 00-07445
An approach for measuring safety training effectiveness
Oberman, George
Occupational Health & Safety v65n12 PP: 48, 58 Dec 1996
ISSN: 0362-4064 JRNL CODE: OHS
WORD COUNT: 1398

...TEXT: of the supervisors or the student's peers; an interview with the supervisor about the participant's **performance before** and **after instruction** ; observations by the supervisor using structured checklists developed from specific training programs; techniques such as job safety...

21/3,K/3 (Item 1 from file: 647)
DIALOG(R)File 647:CMP Computer Fulltext
(c) 2004 CMP Media, LLC. All rts. reserv.

01030634 CMP ACCESSION NUMBER: EET19940919S0052
Hybrid processors herald performance (TechFiles)
ELECTRONIC ENGINEERING TIMES, 1994, n 815, PG58
PUBLICATION DATE: 940919
JOURNAL CODE: EET LANGUAGE: English
RECORD TYPE: Fulltext
SECTION HEADING: Technology
WORD COUNT: 1251

... forwarding within the multiple issues of instructions; and dynamic-execution relocation between the two compute engines.
The **instruction pre - fetch** is **performed** by another four-stage pipeline that is decoupled by a FIFO instruction **buffer** from the dual-execution pipelines. The third stage of this fetch pipeline is dedicated to a table...

21/3,K/4 (Item 1 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

02349943 SUPPLIER NUMBER: 57588385 (USE FORMAT 7 OR 9 FOR FULL TEXT)
PATENT WATCH. (News Briefs)

Belgard, Rich
Microprocessor Report, 13, 15, NA
Nov 15, 1999
ISSN: 0899-9341 LANGUAGE: English RECORD TYPE: Fulltext
WORD COUNT: 682 LINE COUNT: 00061

... is an instruction queue in a scalar RISC processor that may reorder instructions for dispatch to the **execution** units such that a **branch instruction** is dispatched before instructions that precede it in the instruction stream. In this way, the **branch instruction** can fill the I-cache simultaneously with the **execution** of some **prior instructions** .
5,909,587
Multi-chip superscalar microprocessor module
Filed: October 24, 1997
Issued: June 1, 1999
Inventor...

21/3,K/5 (Item 2 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

02189483 SUPPLIER NUMBER: 20841610 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Patent Watch. (Technology Information)
Belgard, Rich
Microprocessor Report, v12, n8, p26(1)
June 22, 1998
ISSN: 0899-9341 LANGUAGE: English RECORD TYPE: Fulltext
WORD COUNT: 574 LINE COUNT: 00053

... every branch instruction, the instruction-fetch stage begins fetching both the branch target and the next sequential **instruction before** the **execution** unit completes processing the **branch instruction** .
5,696,956
Dynamically programmable reduced-instruction-set computer with programmable processor loading on program-number- field...

21/3,K/6 (Item 3 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01879813 SUPPLIER NUMBER: 17854470 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Patent Watch. (News Briefs)
Microprocessor Report, v9, n16, p25(1)
Dec 4, 1995
ISSN: 0899-9341 LANGUAGE: English RECORD TYPE: Fulltext
WORD COUNT: 644 LINE COUNT: 00058

... Seiko Epson
Filed: May 24, 1993
Claims: 3
A method for use in a microprocessor to return **execution** to a main program after processing an interrupt. The method comprises fetching instructions from a **main instruction** stream to a main buffer section of a prefetch buffer and **executing** those instructions. The method also provides for returning to the **main instruction** stream without requiring **prefetching** of **instructions** already fetched. Similarly, the method also provides for handling of nested interrupts.
5,446,912
Partial width...

21/3,K/7 (Item 4 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01869660 SUPPLIER NUMBER: 17711998 (USE FORMAT 7 OR 9 FOR FULL TEXT)
CICS EI-level-2 tracing. (Technology Tutorial)(Tutorial)
Wright, Andrew
Enterprise Systems Journal, v10, n11, p46(5)
Oct, 1995
DOCUMENT TYPE: Tutorial ISSN: 1053-6566 LANGUAGE: English
RECORD TYPE: Fulltext; Abstract
WORD COUNT: 3333 LINE COUNT: 00258

...ABSTRACT: 2 entries contain much more data than level 1. Four areas are traced for each EXEC CICS **command** , **before** and **after execution** . Each trace entry has four associated data items, the first being information required to reconstruct the EXEC...

21/3,K/8 (Item 5 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01504582 SUPPLIER NUMBER: 11954808 (USE FORMAT 7 OR 9 FOR FULL TEXT)
The REI movement: technology may be the key to success for a new initiative to teach special education students in regular classrooms. (Regular Education Initiative) (Guest Editorial) (Column)
Wiener, Roberta
Electronic Learning, v11, n6, p12(1)
March, 1992
DOCUMENT TYPE: Column ISSN: 0278-3258 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 614 LINE COUNT: 00053

... developed as a cooperative learning experience.
Instructional Management. Achievement can be demonstrated by taking samples of student **performance before** and **after instruction** . This helps regular and special students master learning objectives and provides documentation of growth and change. Many...

21/3,K/9 (Item 6 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01466032 SUPPLIER NUMBER: 11485858 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Optimizing assembly language programs, part 2. (Power Programming) (Tutorial)
Duncan, Ray
PC Magazine, v10, n21, p405(4)
Dec 17, 1991
DOCUMENT TYPE: Tutorial ISSN: 0888-8507 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 2628 LINE COUNT: 00218

... reloads the IP. The BIU must then start from scratch to fetch instruction bytes from the new **execution** address and load them into the prefetch queue, while the EU must "stall" until a complete instruction has been retrieved. Then, any memory accesses that are involved in the **execution** of that **first instruction** at the new address will interfere with the BIU's ability to **prefetch** the subsequent **instruction** , and so on. It may be quite a while before the BIU can completely fill the prefetch ...

21/3,K/10 (Item 7 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01404662 SUPPLIER NUMBER: 10618207

The evolution of instruction sequencing. (includes related article on the impact of small recurring delays)
Krick, Robert F.; Dollas, Apostolos
Computer, v24, n4, p5(11)
April, 1991
ISSN: 0018-9162 LANGUAGE: ENGLISH RECORD TYPE: ABSTRACT

...ABSTRACT: memory speed. IS consists of determining the memory address of a required instruction, fetching the instruction and **executing** it. Memory bandwidth, instruction buffers, cache memory, interleaving and the distribution of instructions in memory affect IS **performance** . The use and impacts of each are discussed. Methods of improving IS **performance** include compilation of a program's **instructions prior** to **execution** , the use of conditional **branch instruction** prediction strategies, rearranging instructions and reducing memory conflicts to improve the cache-hit ratio, **execution** of multiple instructions in a single cycle and increasing concurrent **execution** through the use of very long instruction words. AT&T's CRISP microprocessor employs several methods for improving IS **performance** . Future IS approaches are described.

21/3,K/11 (Item 8 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01318076 SUPPLIER NUMBER: 07849122 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Multiple instruction dispatch drives RISC chip to 66 Mips. (Intel's 80960CA microprocessor)
Wilson, Ron
Computer Design, v28, n19, p22(3)
Oct 1, 1989
ISSN: 0010-4566 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 2042 LINE COUNT: 00160

... an integer divide).
Keeping things moving
The chip has a number of sophisticated touches to keep the **execution** rate up. One deals with the old nemesis of RISC architectures, the **branch instruction** . The sequencer attempts to spot branches in the **instruction** stream and **pre - execute** them, completely overlapping the **branch** with other **instructions** .
In the case of conditional branches, the instruction contains a bit to indicate whether the branch is...

21/3,K/12 (Item 9 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01294544 SUPPLIER NUMBER: 07224616 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Investigating the debugging registers of the Intel 386 microprocessor.
Hansen, Marion; Stuecklen, Nick
Microsoft Systems Journal, v4, n3, p39(12)
May, 1989
ISSN: 0889-9932 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 3771 LINE COUNT: 00296

... byte.
Debug Exceptions
Breakpoints set on instructions cause faults; all other debug conditions cause traps. (Faults break **before executing** the **instruction** at the specified address. Traps report a data access breakpoint **after executing** the **instruction** that accesses the given memory item.) The debug exception can report faults and traps at the same...

21/3,K/13 (Item 10 from file: 275)

DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01237570 SUPPLIER NUMBER: 06160604 (USE FORMAT 7 OR 9 FOR FULL TEXT)
PC Tutor. (help column) (column)
Hummel, Robert L.; Dove, Henry; Nichols, Raymond
PC Magazine, v7, n1, p385(4)
Jan 12, 1988
DOCUMENT TYPE: column ISSN: 0888-8507 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT
WORD COUNT: 2067 LINE COUNT: 00151

... is an example of the pre-fetch queue at work.
By the time the CPU begins to **execute** the MOV **instruction** , the **pre - fetch** circuitry has already buffered the next instruction. Although the RAM holding the JMP is modified, the CPU still **executes** the **original instruction** , as read from the pre-fetch queue. When tracing single instructions under DEBUG, the CPU is being used for other tasks (such as, for instance, displaying the DEBUG output) besides **executing** the program. This forces the re-reading of each instruction just before it is **executed** . But, thanks to the G command, the program instructions were **executed** without interruption and the CPU used the queued instructions. Schemes similar to this one are sometimes used...

21/3,K/14 (Item 1 from file: 16)
DIALOG(R)File 16:Gale Group PROMT(R)
(c) 2004 The Gale Group. All rts. reserv.

03605322 Supplier Number: 45076097 (USE FORMAT 7 FOR FULLTEXT)
ULTRASPARC
Electronics Times, p23
Oct 20, 1994
Language: English Record Type: Fulltext
Document Type: Magazine/Journal; Trade
Word Count: 821

... of an otherwise resource-rich processor.
To get round the problem, UltraSparc has a 12-entry input **buffer** to isolate the **instruction pre - fetch** from **execution** .
Up to four **instructions** are fetched per cycle to the **buffer** . When the processor meets a conditional instruction, it must chose which further segment of code to pre...

21/3,K/15 (Item 2 from file: 16)
DIALOG(R)File 16:Gale Group PROMT(R)
(c) 2004 The Gale Group. All rts. reserv.

03559795 Supplier Number: 45001217 (USE FORMAT 7 FOR FULLTEXT)
Hybrid processors herald performance
Electronic Engineering Times, p58
Sept 19, 1994
Language: English Record Type: Fulltext
Document Type: Magazine/Journal; Trade
Word Count: 1261

... forwarding within the multiple issues of instructions; and dynamic-execution relocation between the two compute engines.
The **instruction pre - fetch** is **performed** by another four-stage pipeline that is decoupled by a FIFO instruction **buffer** from the dual-execution pipelines. The third stage of this fetch pipeline is dedicated to a table...

```
Set     Items   Description
S1         46   AU='COPELAND G' OR AU='COPELAND G P' OR AU='COPELAND GEORG-
               E' OR AU='COPELAND GEORGE P' OR AU='COPELAND GEORGE PRENTICE'
S2         38   AU='CONNER M' OR AU='CONNER M H' OR AU='CONNER MICHAEL' OR
               AU='CONNER MICHAEL HADEN'
S3         44   AU='FLURRY G' OR AU='FLURRY G A' OR AU='FLURRY GREGORY ALA-
               IN' OR AU='FLURRY GREGORY ALAN'
S4        110   S1 OR S2 OR S3
S5         92   S4 AND IC=G06F?
S6         24   S5 AND IC=G06F-015?
File 347:JAPIO Nov 1976-2004/Apr(Updated 040802)
         (c) 2004 JPO & JAPIO
File 348:EUROPEAN PATENTS 1978-2004/Jul W04
         (c) 2004 European Patent Office
File 349:PCT FULLTEXT 1979-2002/UB=20040729,UT=20040722
         (c) 2004 WIPO/Univentio
File 350:Derwent WPIX 1963-2004/UD,UM &UP=200449
         (c) 2004  Thomson Derwent
```

6/5/1 **(Item 1 from file: 350)**
DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

016349373 **Image available**
WPI Acc No: 2004-507506/200448
XRPX Acc No: N04-401062

Web service operation invoking method in personal digital assistant, involves retrieving web service implementation based on selection scope associated with web service in received web service operation request

Patent Assignee: INT BUSINESS MACHINES CORP (IBMC); IBM UK LTD (IBMC)
Inventor: **FLURRY G A** ; HOLDSWORTH S A J
Number of Countries: 106 Number of Patents: 002
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
WO 200455693	A1	20040701	WO 2003GB4970	A	20031117	200448	B
US 20040139151	A1	20040715	US 2002322053	A	20021217	200448	

Priority Applications (No Type Date): US 2002322053 A 20021217
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
WO 200455693	A1	E	33	G06F-017/30	
US 20040139151	A1			G06F-015/16	

Designated States (National): AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CO CR CU CZ DE DK DM DZ EC EE EG ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NI NO NZ OM PG PH PL PT RO RU SC SD SE SG SK SL SY TJ TM TN TR TT TZ UA UG UZ VC VN YU ZA ZM ZW
Designated States (Regional): AT BE BG CH CY CZ DE DK EA EE ES FI FR GB GH GM GR HU IE IT KE LS LU MC MW MZ NL OA PT RO SD SE SI SK SL SZ TR TZ UG ZM ZW

Abstract (Basic): WO 200455693 A1

NOVELTY - A selection scope associated with web service in a received web service operation request is determined from several selection scopes. A web service implementation is retrieved based on the determined selection scope and accordingly the web service operation is provided.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:

(1) web service operation invoking apparatus;

(2) web service operation invoking program.

USE - For providing web service operation in distributed data processing system e.g. personal digital assistant (PDA), notebook computer, handheld computer, kiosk and web appliance in network such as intranet, local area network, and wide area network for business application.

ADVANTAGE - The web service implementation is identified reliably and the corresponding web service is accessed accurately.

DESCRIPTION OF DRAWING(S) - The figure shows the flowchart explaining the web service operation invoking process.

pp; 33 DwgNo 7/8

Title Terms: WEB; SERVICE; OPERATE; INVOKE; METHOD; PERSON; DIGITAL; ASSIST ; RETRIEVAL; WEB; SERVICE; IMPLEMENT; BASED; SELECT; SCOPE; ASSOCIATE; WEB; SERVICE; RECEIVE; WEB; SERVICE; OPERATE; REQUEST
Derwent Class: T01
International Patent Class (Main): **G06F-015/16** ; **G06F-017/30**
File Segment: EPI

6/5/2 **(Item 2 from file: 350)**
DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

016340537 **Image available**
WPI Acc No: 2004-498434/200447
XRPX Acc No: N04-393685

Web service deployment method in data processing system, involves retrieving deployment descriptor that describes web service deployment

Patent Assignee: IBM CORP (IBMC)
Inventor: BERKLAND P T; CURTIS B A; **FLURRY G A**
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 20040117425	A1	20040617	US 2002322051	A	20021217	200447	B

Priority Applications (No Type Date): US 2002322051 A 20021217
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 20040117425	A1		25	G06F-015/16	

Abstract (Basic): US 20040117425 A1

NOVELTY - The method involves retrieving a deployment descriptor that describes how the web service is deployed in the data processing system. The location type for the web service, is determined from the deployment descriptor. The web service is deployed based on the determined location type.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:

(1) web service deployment program; and

(2) web service bus off-ramp.

USE - For deploying web service in data processing system e.g. personal digital assistant (PDA), notebook computer, handheld computer, kiosk, web appliance, using web service description language (WSDL) document, JavaBean.

ADVANTAGE - Client is allowed to continue to use the data type name information in the existing WSDL, by creating description of service and by using both WSDL and JavaBean.

DESCRIPTION OF DRAWING(S) - The figure shows the flowchart explaining the web service deployment method.

pp; 25 DwgNo 6/15

Title Terms: WEB; SERVICE; DEPLOY; METHOD; DATA; PROCESS; SYSTEM; RETRIEVAL ; DEPLOY; DESCRIBE; DESCRIBE; WEB; SERVICE; DEPLOY
Derwent Class: T01
International Patent Class (Main): **G06F-015/16**
File Segment: EPI

6/5/3 (Item 3 from file: 350)

DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

016320836 **Image available**
WPI Acc No: 2004-478731/200445
XRPX Acc No: N04-377427

Web service selection method in web service innovation framework, involves generating service object based on selected candidate for web service implementation, and providing generated object to client for accessing web service

Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)
Inventor: BERKLAND P T; **FLURRY G A** ; HOLDSWORTH S A J; HUTCHISON E A
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 20040111525	A1	20040610	US 2002314813	A	20021209	200445	B

Priority Applications (No Type Date): US 2002314813 A 20021209
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 20040111525	A1		14	G06F-015/16	

Abstract (Basic): US 20040111525 A1

NOVELTY - A web service request including identifier associated with port Type, is received from client terminal. A list of candidate

for web service implementations is compiled based on the identifier, and one of the candidate in list is selected based on specific selection criteria. A service object is generated based on the selected candidate, and generated object is provided to client for accessing the web service.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:

(1) computer program product comprising computer readable medium stored with web service selection program; and

(2) web service selection apparatus.

USE - For selecting web services provided to client, in network data processing system e.g. web service innovation framework (WSIF) connected to network such as local area network (LAN) and wide area network (WAN).

ADVANTAGE - Enables to dynamically discover and select the web services for implementation in run time.

DESCRIPTION OF DRAWING(S) - The figure shows the block diagram of the WSIF.

pp; 14 DwgNo 4/10

Title Terms: WEB; SERVICE; SELECT; METHOD; WEB; SERVICE; FRAMEWORK; GENERATE; SERVICE; OBJECT; BASED; SELECT; CANDIDATE; WEB; SERVICE; IMPLEMENT; GENERATE; OBJECT; CLIENT; ACCESS; WEB; SERVICE

Derwent Class: T01

International Patent Class (Main): **G06F-015/16**

File Segment: EPI

6/5/4 (Item 4 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2004 Thomson Derwent. All rts. reserv.

016196666 **Image available**

WPI Acc No: 2004-354552/200433

XRPX Acc No: N04-283240

Document object model processing method for verifying context between extensible mark-up language tags, involves replacing current processing element in document object model with placeholder element

Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)

Inventor: CLAUSSEN C S; **FLURRY G A** ; MCCLAIN M D; XU L; ZUMBRUNNEN B C

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 6718516	B1	20040406	US 99409373	A	19990930	200433	B

Priority Applications (No Type Date): US 99409373 A 19990930

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 6718516	B1		25	G06F-015/00	

Abstract (Basic): US 6718516 B1

NOVELTY - A current processing element in the document object model (DOM) is replaced with a placeholder element having attributes indicating state of the current processing element. A state information is loaded from a set of element, and is processed by a clean-up element that is added to the DOM.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:

(1) extensible markup language (XML) DOM tree processing method; and

(2) computer program product for processing DOM.

USE - For processing document object model (DOM) for verifying context between multiple related custom tags such as extensible mark-up language (XML) tags and hypertext markup language (HTML) tags using personal computer, notebook computer, Internet appliance or pervasive computing device such as personal digital assistant (PDA)and palm computer.

ADVANTAGE - Enables effective processing of DOM and verification of

context between multiple custom tags by the clean-up element, thereby eliminating unauthorized viewing of the codes.

DESCRIPTION OF DRAWING(S) - The figure shows the flowchart explaining the servlet generation process.

pp; 25 DwgNo 2/11

Title Terms: DOCUMENT; OBJECT; MODEL; PROCESS; METHOD; VERIFICATION; CONTEXT; EXTEND; MARK; UP; LANGUAGE; TAG; REPLACE; CURRENT; PROCESS; ELEMENT; DOCUMENT; OBJECT; MODEL; ELEMENT

Derwent Class: P85; T01

International Patent Class (Main): **G06F-015/00**

International Patent Class (Additional): G09G-005/00

File Segment: EPI; EngPI

6/5/5 (Item 5 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2004 Thomson Derwent. All rts. reserv.

016187358 **Image available**

WPI Acc No: 2004-345244/200432

XRPX Acc No: N04-275924

Dynamic data table generation method in server for Internet publishing applications, involves formatting dynamic data into table by modifying markup language in document, based on instantiated table format object

Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)

Inventor: **CONNER M H** ; MCCLAIN M D; XU L

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 6718515	B1	20040406	US 99455711	A	19991207	200432	B

Priority Applications (No Type Date): US 99455711 A 19991207

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 6718515	B1		19	G06F-015/00	

Abstract (Basic): US 6718515 B1

NOVELTY - A table format object is instantiated with dynamic data that is extracted from a data object through a pluggable interface that is customized with a developer supplied data getter object to access a custom data-object, in response to a client's request. Dynamic data is formatted into a table by modifying markup language in the document, based on the table format object.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:

(1) computer program product in computer readable medium for dynamic data table generation;

(2) server.

USE - For generating dynamic data table in server (claimed) for writing server page using Java server page (JSP), for Internet publishing applications.

ADVANTAGE - Performs correct and effective generation of dynamic data tables that are widely used for writing server pages.

DESCRIPTION OF DRAWING(S) - The figure shows a flowchart illustrating JSP page handling process.

pp; 19 DwgNo 2/12

Title Terms: DYNAMIC; DATA; TABLE; GENERATE; METHOD; SERVE; PUBLICATION; APPLY; FORMAT; DYNAMIC; DATA; TABLE; MODIFIED; LANGUAGE; DOCUMENT; BASED; TABLE; FORMAT; OBJECT

Derwent Class: T01

International Patent Class (Main): **G06F-015/00**

File Segment: EPI

6/5/6 (Item 6 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2004 Thomson Derwent. All rts. reserv.

016157587 **Image available**
WPI Acc No: 2004-315474/200429
XRPX Acc No: N04-251401

Distributed web application support system for electronic business applications, has server having common cache which has different areas for storing code and data associated with commands and Java server pages requested by client

Patent Assignee: CONNER M H (CONN-I); COPELAND G P (COPE-I); FLURRY G A (FLUR-I)
Inventor: **CONNER M H ; COPELAND G P ; FLURRY G A**
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
US 20040073630	A1	20040415	US 2000740403	A	20001218	200429 B

Priority Applications (No Type Date): US 2000740403 A 20001218
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 20040073630	A1		16	G06F-015/16	

Abstract (Basic): US 20040073630 A1

NOVELTY - The system has a server having a common cache which has an area for storing the code and the data associated with the commands requested by client terminal and another area for storing code and data associated with the Java server pages (JSP) requested by the client. The cache also has a common area for storing the code and data which are common to both commands and the server pages.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:

(1) executable code and data storage method;

(2) computer program product comprising recorded medium storing executable code and data storage program; and

(3) executable code and data storage system.

USE - For supporting distributed web application between client and server, in electronic business applications.

ADVANTAGE - Improves the responsiveness of distributed web applications between clients and servers. Enhances the cache efficiency by allowing the commands and JSPs to be combined in the same cache. The frequency of database access is decreased and the data dependency tracking logic is enhanced. The cached content is delivered to the client at a high speed.

DESCRIPTION OF DRAWING(S) - The figure shows a schematic view explaining the data granularity in a webpage.

dynamic content fragments (50)
dynamic content data (52)
product data command (66)
shopper data command (68)
shopping cart data command (70)
pp; 16 DwgNo 2/5

Title Terms: DISTRIBUTE; WEB; APPLY; SUPPORT; SYSTEM; ELECTRONIC; BUSINESS; APPLY; SERVE; COMMON; CACHE; AREA; STORAGE; CODE; DATA; ASSOCIATE; COMMAND; SERVE; PAGE; REQUEST; CLIENT
Derwent Class: T01; T05
International Patent Class (Main): **G06F-015/16**
File Segment: EPI

6/5/7 (Item 7 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

015769526 **Image available**
WPI Acc No: 2003-831728/200377
XRPX Acc No: N03-664683

Role specific object processing method in data processing system, involves retrieving data from cache, on receiving request from client

which specifies uniform resource identifier and associated cookie value of stored data

Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)
Inventor: AGARWALLA R S; CHALLENGER J R H; **COPELAND G P** ; IYENGAR A K; LINEHAN M H; MEDURI S
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 20030191812	A1	20031009	US 200134724	A	20011219	200377	B

Priority Applications (No Type Date): US 200134724 A 20011219
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 20030191812	A1		68	G06F-015/16	

Abstract (Basic): US 20030191812 A1

NOVELTY - The method involves storing data contents in cache memory, corresponding to specific uniform resource identifier (URI) along with associated cookie value. The stored contents are retrieved on receiving a request containing the URI and the cooker value of the data, from the client.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are included for the following:

(1) object processing apparatus; and

(2) computer program product for data processing.

USE - For processing role specific object in data processing system.

ADVANTAGE - The cache memory improves responsiveness of web site containing dynamic web content.

DESCRIPTION OF DRAWING(S) - The figure shows the block diagram of the cache management unit within a computing device.

pp; 68 DwgNo 6A/11

Title Terms: ROLE; SPECIFIC; OBJECT; PROCESS; METHOD; DATA; PROCESS; SYSTEM ; RETRIEVAL; DATA; CACHE; RECEIVE; REQUEST; CLIENT; SPECIFIED; UNIFORM; RESOURCE; IDENTIFY; ASSOCIATE; COOKIE; VALUE; STORAGE; DATA
Derwent Class: T01
International Patent Class (Main): **G06F-015/16**
File Segment: EPI

6/5/8 (Item 8 from file: 350)

DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

015769521 **Image available**
WPI Acc No: 2003-831723/200377
XRPX Acc No: N03-664678

Object processing method in distributed data processing system, involves generating link elements in response to determination of expansion attribute in specific link element of top-level object

Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)
Inventor: CHALLENGER J R H; **CONNER M H** ; **COPELAND G P** ; IYENGAR A K
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 20030191800	A1	20031009	US 200134771	A	20011219	200377	B

Priority Applications (No Type Date): US 200134771 A 20011219
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 20030191800	A1		69	G06F-015/16	

Abstract (Basic): US 20030191800 A1

NOVELTY - A message containing headers and payload portion with top-level object comprising specific link element to next level object, is received at computing device (600). A set of link elements is generated in response to determination that specific link element

comprises an expansion attribute, and in accordance with the parameter associated with the attribute.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:

(1) object processing apparatus;
(2) data structure; and
(3) computer program product for processing object.

USE - For processing objects in distributed data processing system.

ADVANTAGE - Allows a cache management unit of the computing device to operate without regard to whether the computing device acts as a server, client or hub located throughout the network.

DESCRIPTION OF DRAWING(S) - The figure shows the block diagram of the cache management unit of the computing device.

computing device (600)
cache management unit (602)
object database (604)
cache ID database (612)
dependency database (618)
pp; 69 DwgNo 6A/11

Title Terms: OBJECT; PROCESS; METHOD; DISTRIBUTE; DATA; PROCESS; SYSTEM; GENERATE; LINK; ELEMENT; RESPOND; DETERMINE; EXPAND; ATTRIBUTE; SPECIFIC; LINK; ELEMENT; TOP; LEVEL; OBJECT

Derwent Class: T01

International Patent Class (Main): **G06F-015/16**

File Segment: EPI

6/5/9 (Item 9 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2004 Thomson Derwent. All rts. reserv.

015685594 **Image available**

WPI Acc No: 2003-747783/200370

XRPX Acc No: N03-599464

Processing method for fragment caching in Internet, involves sending request including source identifiers, for receiving fragment set, when set of fragments associated with set of source identifiers are not in cache

Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)

Inventor: CHALLENGER J R H; **COPELAND G P** ; IYENGAR A K; LINEHAN M H

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 20030188021	A1	20031002	US 200134726	A	20011219	200370	B

Priority Applications (No Type Date): US 200134726 A 20011219

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 20030188021	A1		68	G06F-015/16	

Abstract (Basic): US 20030188021 A1

NOVELTY - A request message with a set of source identifiers is transmitted, when a set of fragments associated with a set of source identifiers are not present in cache. A response message with a set of fragments is received based on the request.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:

(1) an apparatus for processing objects; and
(2) the computer program product of object processing

USE - For fragment caching of data objects in Internet.

ADVANTAGE - By using the cache, large amount of high speed memory usage is replaced and thus improves responsiveness of web site. Due to the provision of compression by fragment cache, size and cost are reduced and performance is improved.

DESCRIPTION OF DRAWING(S) - The figure shows the block diagram of cache management unit for fragment-supporting cache.

computing device (600)
fragment-supporting cache management unit (602)
cache ID database (612)
cache ID (614)
cache index (616)
pp; 68 DwgNo 6A/11
Title Terms: PROCESS; METHOD; FRAGMENT; SEND; REQUEST; SOURCE; IDENTIFY; RECEIVE; FRAGMENT; SET; SET; FRAGMENT; ASSOCIATE; SET; SOURCE; IDENTIFY; CACHE
Derwent Class: T01
International Patent Class (Main): **G06F-015/16**
File Segment: EPI

6/5/10 (Item 10 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

015685593 **Image available**
WPI Acc No: 2003-747782/200370
XRPX Acc No: N03-599463
Data object processing method for fragment caching in Internet, involves determining caching of fragment by computing device that has fragment-supporting cache management unit, after receiving response for fragment request
Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)
Inventor: AGARWALLA R S; CHALLENGER J R H; **COPELAND G P** ; IYENGAR A K; MEDURI S
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 20030188016	A1	20031002	US 200134770	A	20011219	200370	B

Priority Applications (No Type Date): US 200134770 A 20011219
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 20030188016	A1		68	G06F-015/173	

Abstract (Basic): US 20030188016 A1
NOVELTY - A computing device receives request message with source identifier, for fragment and determines whether the request message is processed by another computing device. The fragment caching by the another device having the fragment- supporting cache management unit is determined after the response message is received.
DETAILED DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:
(1) an apparatus for processing objects; and
(2) a computer program product for data object processing
USE - For object data processing by fragment caching, in Internet.
ADVANTAGE - Provides distributed fragment caching mechanism by the cache management unit, thus reduces cache size provided by the fragment compression. Hence cost is reduced and performance is improved.
DESCRIPTION OF DRAWING(S) - The figure shows the block diagram of typical web page composed of fragments.
dynamic content fragments (200)
product display Java server page (JSP) (204)
product detail display (208)
personalized greeting (210)
abbreviated shopping CART JSP (212)
pp; 68 DwgNo 2/11
Title Terms: DATA; OBJECT; PROCESS; METHOD; FRAGMENT; DETERMINE; FRAGMENT; COMPUTATION; DEVICE; FRAGMENT; SUPPORT; CACHE; MANAGEMENT; UNIT; AFTER; RECEIVE; RESPOND; FRAGMENT; REQUEST
Derwent Class: T01
International Patent Class (Main): **G06F-015/173**
File Segment: EPI

6/5/11 (Item 11 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

015685591 **Image available**
WPI Acc No: 2003-747780/200370
XRPX Acc No: N03-599461
Data processing method in computer network, involves retrieving message header from message, that indicates that message main portion includes linking element to next-level fragment
Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)
Inventor: AGARWALLA R S; CHALLENGER J R H; **COPELAND G P** ; IYENGAR A K; LINEHAN M H
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 20030188009	A1	20031002	US 200134748	A	20011219	200370	B

Priority Applications (No Type Date): US 200134748 A 20011219
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 20030188009	A1		67	G06F-015/16	

Abstract (Basic): US 20030188009 A1
NOVELTY - A hypertext transport protocol (HTTP) response message having a set of message headers and a message main portion with a top-level fragment, is received at a computing device e.g. personal digital assistant. The message header is retrieved from the message to indicate that the message main portion includes a linking element to a next-level fragment.
DETAILED DESCRIPTION - INDEPENDENT CLAIMS are included for the following:
(1) data processing apparatus;
(2) computer program product for processing data object; and
(3) data structure.
USE - In data processing system (claimed) comprising computing devices such as mainframe, personal computer, personal digital assistant (PDA), and wireless phone connected to computer network such as Internet, local area network (LAN), wireless LAN, and wide area network (WAN).
ADVANTAGE - Permits distributed caching technique resulting in reduced fragment cache sizes that are maintained by lightweight process in standard manner through Internet with minimal regard to cache location.
DESCRIPTION OF DRAWING(S) - The figure shows a flowchart explaining data processing procedure.
pp; 67 DwgNo 6B/11
Title Terms: DATA; PROCESS; METHOD; COMPUTER; NETWORK; RETRIEVAL; MESSAGE; HEADER; MESSAGE; INDICATE; MESSAGE; MAIN; PORTION; LINK; ELEMENT; LEVEL; FRAGMENT
Derwent Class: T01
International Patent Class (Main): **G06F-015/16**
File Segment: EPI

6/5/12 (Item 12 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

015681130 **Image available**
WPI Acc No: 2003-743319/200370
XRPX Acc No: N03-595208
Cache coordinating method for web application server address spaces, involves sending information from storage device to coordinating address space in response to absence of information in coordinating address space
Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)

Inventor: **COPELAND G P** ; MCCLAIN M D
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 6615235	B1	20030902	US 99359276	A	19990722	200370	B

Priority Applications (No Type Date): US 99359276 A 19990722
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 6615235	B1		27	G06F-015/16	

Abstract (Basic): US 6615235 B1

NOVELTY - The method involves sending a request to a coordinating address space in response to an absence of information for the page in a cache within a web application server (206,208) address space. A storage device associated with a data processing system is accessed for the information and the information is sent to the coordinating address space in response to an absence of information in coordinating address space.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:

(a) a data processing system for coordinating caches for a set of web application server address spaces

(b) a computer program product in a computer readable medium for coordinating data in a set of web application server address spaces.

USE - Used for coordinating caches in web application server address spaces.

ADVANTAGE - The method allows caching and handling of dynamic contents. The method increases the access rate and decreases the update rate.

DESCRIPTION OF DRAWING(S) - The drawing shows a block diagram of a server system.

Server system (200)
Web server nodes (202,204)
Web application servers (206,208)
Sprayer (210)
Database (212)
Interconnect (214)
Storage devices (216,218,220,222)
pp; 27 DwgNo 2/17

Title Terms: CACHE; COORDINATE; METHOD; WEB; APPLY; SERVE; ADDRESS; SPACE; SEND; INFORMATION; STORAGE; DEVICE; COORDINATE; ADDRESS; SPACE; RESPOND; ABSENCE; INFORMATION; COORDINATE; ADDRESS; SPACE
Derwent Class: T01
International Patent Class (Main): **G06F-015/16**
File Segment: EPI

6/5/13 (Item 13 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

015668297 **Image available**
WPI Acc No: 2003-730484/200369
XRPX Acc No: N03-583869

Peer nodes communicating method, involves communicating discovery command from current peer node to neighbor peer node and receiving aggregated list of peer nodes at current peer node

Patent Assignee: INT BUSINESS MACHINES CORP (IBMC); IBM UK LTD (IBMC)
Inventor: BECKER C; **CONNER M** ; BECKER C H; **CONNER M H**
Number of Countries: 101 Number of Patents: 003
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 20030131129	A1	20030710	US 200244997	A	20020110	200369	B
WO 200358917	A1	20030717	WO 2003GB53	A	20030109	200369	
AU 2003201646	A1	20030724	AU 2003201646	A	20030109	200421	

Priority Applications (No Type Date): US 200244997 A 20020110
Patent Details:
Patent No Kind Lan Pg Main IPC Filing Notes
US 20030131129 A1 21 G06F-015/173
WO 200358917 A1 E H04L-029/06
Designated States (National): AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CO CR CU CZ DE DK DM DZ EC EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NO NZ OM PH PL PT RO RU SD SE SG SK SL TJ TM TN TR TT TZ UA UG UZ VC VN YU ZA ZM ZW
Designated States (Regional): AT BE BG CH CY CZ DE DK EA EE ES FI FR GB GH GM GR HU IE IT KE LS LU MC MW MZ NL OA PT SD SE SI SK SL SZ TR TZ UG ZM ZW
AU 2003201646 A1 H04L-029/06 Based on patent WO 200358917

Abstract (Basic): US 20030131129 A1
NOVELTY - The method involves communicating a discovery command from a current peer node to one neighbor node that is in communication with the current peer node. An aggregated list of peer nodes is received at the current peer node, in which the aggregated list of the peer nodes includes information about one peer node in communication with the neighbor node.
DETAILED DESCRIPTION - An INDEPENDENT CLAIM is also included for the following
(a) a computer program product in a computer usable medium for communicating among multiple of peer nodes in a network environment
(b) a system for communicating among multiple of peer nodes in a network environment.
USE - User for communicating and distributing information e.g. files and search commands in a network environment.
ADVANTAGE - The method provides efficient communication of information between target devices, reduces network and processor overhead, and also improves performance.
DESCRIPTION OF DRAWING(S) - The drawing shows a flow diagram of a method of communicating among multiple target devices in a network environment.
pp; 21 DwgNo 4/8
Title Terms: PEER; NODE; COMMUNICATE; METHOD; COMMUNICATE; DISCOVER; COMMAND; CURRENT; PEER; NODE; NEIGHBOURING; PEER; NODE; RECEIVE; AGGREGATE; LIST; PEER; NODE; CURRENT; PEER; NODE
Derwent Class: T01; W01
International Patent Class (Main): **G06F-015/173** ; H04L-029/06
International Patent Class (Additional): **G06F-015/16**
File Segment: EPI

6/5/14 (Item 14 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

015452000 **Image available**
WPI Acc No: 2003-514142/200348
XRPX Acc No: N03-408078
Method for processing objects at data processing system in network by receiving first message at computing device and determining that its header indicates that it relates to fragment
Patent Assignee: INT BUSINESS MACHINES CORP (IBMC); IBM UK LTD (IBMC)
Inventor: AGARWALLA R; CHALLENGER J; **COPELAND G** ; **IYENGAR A** ; LINEHAN M; MEDURI S; AGARWALLA R S; CHALLENGER J R H; **COPELAND G P** ; **IYENGAR A K** ; LINEHAN M H
Number of Countries: 101 Number of Patents: 003
Patent Family:
Patent No Kind Date Applicat No Kind Date Week
WO 200353023 A1 20030626 WO 2002GB5712 A 20021218 200348 B
US 20030187935 A1 20031002 US 200134772 A 20011219 200365
AU 2002350971 A1 20030630 AU 2002350971 A 20021218 200420

Priority Applications (No Type Date): US 200134772 A 20011219
Patent Details:
Patent No Kind Lan Pg Main IPC Filing Notes
WO 200353023 A1 E 136 H04L-029/06
Designated States (National): AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CO CR CU CZ DE DK DM DZ EC EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NO NZ OM PH PL PT RO RU SD SE SG SK SL TJ TM TN TR TT TZ UA UG UZ VC VN YU ZA ZM ZW
Designated States (Regional): AT BE BG CH CY CZ DE DK EA EE ES FI FR GB GH GM GR IE IT KE LS LU MC MW MZ NL OA PT SD SE SI SK SL SZ TR TZ UG ZM ZW
US 20030187935 A1 G06F-015/16
AU 2002350971 A1 H04L-029/06 Based on patent WO 200353023

Abstract (Basic): WO 200353023 A1
NOVELTY - A first message is received at a computing device with determining that its header indicates that the first message relates to a cacheable fragment. The latter from the first message is stored in a cache maintained by a cache management unit within the computing device. The cache management unit operates equivalently in support of fragment caching operations in a client, a server, or a hub located throughout the network.
DETAILED DESCRIPTION - INDEPENDENT CLAIMS are included for:
(a) an apparatus for processing objects at data processing system in a network
(b) a computer program product
(c) a data structure for use by a computing display in defining a message that is transmitted on a network
USE - In a data processing system for caching data objects within a computer with an implementation of the distributed caching technique.
ADVANTAGE - Results in reduced fragment cache sizes that are maintainable by lightweight processes in a standard manner throughout the Internet with minimal regard to cache location. An implementation of the distributed caching technique is interoperable with other systems that have not implemented the distributed caching technique.
DESCRIPTION OF DRAWING(S) - The drawing is a flowchart that depicts a fragment supporting cache management unit when processing response messages that contain fragments.
pp; 136 DwgNo 6b/11
Title Terms: METHOD; PROCESS; OBJECT; DATA; PROCESS; SYSTEM; NETWORK; RECEIVE; FIRST; MESSAGE; COMPUTATION; DEVICE; DETERMINE; HEADER; INDICATE; RELATED; FRAGMENT
Derwent Class: T01; W01
International Patent Class (Main): **G06F-015/16** ; H04L-029/06
File Segment: EPI

6/5/15 (Item 15 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

014955812 **Image available**
WPI Acc No: 2003-016326/200301
XRPX Acc No: N03-012246
Software system for supporting distributed web applications, has link associated with call to child server page to execute its corresponding codes stored in cache
Patent Assignee: CONNER M H (CONN-I); COPELAND G P (COPE-I); FLURRY G A (FLUR-I)
Inventor: **CONNER M H** ; **COPELAND G P** ; **FLURRY G A**
Number of Countries: 001 Number of Patents: 001
Patent Family:
Patent No Kind Date Applicat No Kind Date Week
US 20020111992 A1 20020815 US 2000740460 A 20001218 200301 B

Priority Applications (No Type Date): US 2000740460 A 20001218

Patent Details:
Patent No Kind Lan Pg Main IPC Filing Notes
US 20020111992 A1 16 G06F-015/16

Abstract (Basic): US 20020111992 A1
NOVELTY - A cache (150) stores the code for the parent and child server pages. The parent server page does not contain all the codes of the called server page. A link is associated with the call to the child server page to execute the codes of the child server page.
DETAILED DESCRIPTION - INDEPENDENT CLAIMS are included for the following:
(1) Computer program product comprising readable medium storing parent and child server page caching program;
(2) Parent and child server page caching method;
(3) Server; and
(4) Computer product comprising web server and software system.
USE - For supporting distributed web applications such as e-commerce Internet.
ADVANTAGE - Shortens average traffic response time to user requests, without using additional servers. Enhances cache efficiency, as the commands and JSPs are stored in same cache.
DESCRIPTION OF DRAWING(S) - The figure illustrates the partitioning of a common cache.
Cache (150)
pp; 16 DwgNo 4/5
Title Terms: SOFTWARE; SYSTEM; SUPPORT; DISTRIBUTE; WEB; APPLY; LINK; ASSOCIATE; CALL; CHILD; SERVE; PAGE; EXECUTE; CORRESPOND; CODE; STORAGE; CACHE
Derwent Class: T01
International Patent Class (Main): **G06F-015/16**
File Segment: EPI

6/5/16 (Item 16 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

014938623 **Image available**
WPI Acc No: 2002-759332/200282
XRPX Acc No: N02-597886
Distributed web application supporting software system for Internet, has pre-execution and post-execution instruction sequences for executing primary and secondary commands and storing executed commands in cache
Patent Assignee: CONNER M H (CONN-I); COPELAND G P (COPE-I); FLURRY G A (FLUR-I)
Inventor: **CONNER M H ; COPELAND G P ; FLURRY G A**
Number of Countries: 001 Number of Patents: 001
Patent Family:
Patent No Kind Date Applicat No Kind Date Week
US 20020116448 A1 20020822 US 2000740399 A 20001218 200282 B

Priority Applications (No Type Date): US 2000740399 A 20001218
Patent Details:
Patent No Kind Lan Pg Main IPC Filing Notes
US 20020116448 A1 16 G06F-015/16

Abstract (Basic): US 20020116448 A1
NOVELTY - The system has a pre-execution instruction sequence which is invoked after the execution of a primary command in order to execute several secondary commands and return all the executed commands to the client. A post-execution instruction sequence is invoked after the execution of the primary command to store the secondary commands in a cache along with the primary command.
DETAILED DESCRIPTION - INDEPENDENT CLAIMS are included for the following:
(1) Method of cofetching several commands in an object oriented software system;

(2) commands cofetching system;
(3) computer program product for cofetching commands; and
(4) Computer product comprising the software system.
USE - For supporting distributed web applications between client and server connected to network such as Internet.
ADVANTAGE - Improves server responsiveness by avoiding the need to issue separate request for every command. Avoids the need for accurate anticipation of the server-to-command ratio and thus optimizes the use of the cache.
DESCRIPTION OF DRAWING(S) - The figure shows the fragment and data granularity for a web site.
pp; 16 DwgNo 2/5
Title Terms: DISTRIBUTE; WEB; APPLY; SUPPORT; SOFTWARE; SYSTEM; PRE; EXECUTE; POST; EXECUTE; INSTRUCTION; SEQUENCE; EXECUTE; PRIMARY; SECONDARY; COMMAND; STORAGE; EXECUTE; COMMAND; CACHE
Derwent Class: T01
International Patent Class (Main): **G06F-015/16**
File Segment: EPI

6/5/17 (Item 17 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

014861723 **Image available**
WPI Acc No: 2002-682429/200273
XRPX Acc No: N02-538819
Distributed computer system for web applications, detects affinity break between client and server when generation ID included in request from client, differs from generation ID recorded by server
Patent Assignee: CONNER M H (CONN-I); COPELAND G P (COPE-I); FLURRY G A (FLUR-I)
Inventor: **CONNER M H ; COPELAND G P ; FLURRY G A**
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 20020116474	A1	20020822	US 2000740531	A	20001218	200273	B

Priority Applications (No Type Date): US 2000740531 A 20001218
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 20020116474	A1		16	G06F-015/16	

Abstract (Basic): US 20020116474 A1
NOVELTY - Each server (14a-14c) increments and stores a numerical valued generation ID included in each request from clients (20a-20c). An affinity break between a client and a server, is detected when the received generation ID differs from the generation ID recorded by the server.
DETAILED DESCRIPTION - INDEPENDENT CLAIMS are included for the following:
(1) Affinity break detection method;
(2) Computer program product;
(3) Computer program product storing instructions to execute affinity break detection; and
(4) Server.
USE - Used for web applications such as e-commerce.
ADVANTAGE - The responsiveness of distributed web applications between client and server, is improved by caching the dynamic content, and the average response time to process client requests, is shortened without any costly investment in the server.
DESCRIPTION OF DRAWING(S) - The figure shows the block diagram of client server hierarchy for a typical web site.
Servers (14a-14c)
Clients (20a-20c)
pp; 16 DwgNo 1/5
Title Terms: DISTRIBUTE; COMPUTER; SYSTEM; WEB; APPLY; DETECT; AFFINITY;

BREAK; CLIENT; SERVE; GENERATE; ID; REQUEST; CLIENT; DIFFER; GENERATE; ID ; RECORD; SERVE
Derwent Class: T01; W01
International Patent Class (Main): **G06F-015/16**
File Segment: EPI

6/5/18 (Item 18 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

014317525 **Image available**
WPI Acc No: 2002-138227/200218
XRPX Acc No: N02-104080
Applet communication method for information retrieval on computer networks using interconnected Java applets to share data
Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)
Inventor: **CONNER M H** ; CURTIS B A; HSU J M
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 6314448	B1	20011106	US 9890838	A	19980604	200218	B

Priority Applications (No Type Date): US 9890838 A 19980604
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 6314448	B1		16	G06F-015/16	

Abstract (Basic): US 6314448 B
NOVELTY - The applet establish an InfoBus communication link with other applets on the network, allowing the applet to share data with other applets during execution. Gateways are established within each local network to allow data sharing between applets.
DETAILED DESCRIPTION - Independent claims are included for
(1) an applet communication system.
(2) an applet communication computer program.
USE - Information searching on computer networks e.g. the Internet, intranet etc.
ADVANTAGE - As the applets share data across the communication network i.e. the Internet , vast amounts of data can be processed and transferred more efficiently, thus decreasing download time and promoting increased user friendly Internet software.
DESCRIPTION OF DRAWING(S) - The drawing shows a flow diagram of the applet execution process.
Dwg.7/7
Title Terms: COMMUNICATE; METHOD; INFORMATION; RETRIEVAL; COMPUTER; NETWORK ; INTERCONNECT; SHARE; DATA
Derwent Class: T01
International Patent Class (Main): **G06F-015/16**
International Patent Class (Additional): **G06F-015/177**
File Segment: EPI

6/5/19 (Item 19 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

014014367 **Image available**
WPI Acc No: 2001-498581/200155
XRPX Acc No: N01-369545
Multi-scripting language supporting method in web page compilation, involves examining document object mode to locate nodes that identify specific scripting language code block
Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)
Inventor: CLAUSSEN C S; **CONNER M H** ; MCCLAIN M D; ZUMBRUNNEN B C
Number of Countries: 002 Number of Patents: 002
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
GB 2357864	A	20010704	GB 200020153	A	20000817	200155	B
US 6732330	B1	20040504	US 99409372	A	19990930	200430	

Priority Applications (No Type Date): US 99409372 A 19990930
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
GB 2357864	A		49	G06F-017/30	
US 6732330	B1			G06F-015/00	

Abstract (Basic): GB 2357864 A

NOVELTY - The start and end of each scripting language code block corresponding to a web page are marked. The web page is complied into extensible markup language (XML) document object model (DOM) to locate nodes that identify a specific code block. The DOM is adjusted to account for script code within the identified block.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:

(a) Web page;

(b) Web page compilation method;

(c) Computer program product for web page compilation

USE - For supporting multi-scripting language in single web page for compiling web page into extensible markup language (XML) document object model (DOM) in client-server environment.

ADVANTAGE - Eases code handling, thereby reducing number of errors. Enables immediately checking the codes for language syntax error.

DESCRIPTION OF DRAWING(S) - The figure shows the client-server environment in which web page compilation method is implemented.

pp; 49 DwgNo 1/11

Title Terms: MULTI; LANGUAGE; SUPPORT; METHOD; WEB; PAGE; COMPILE; DOCUMENT ; OBJECT; MODE; LOCATE; NODE; IDENTIFY; SPECIFIC; LANGUAGE; CODE; BLOCK
Derwent Class: T01
International Patent Class (Main): **G06F-015/00 ; G06F-017/30**
File Segment: EPI

6/5/20 (Item 20 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

013869331 **Image available**
WPI Acc No: 2001-353543/200137
XRPX Acc No: N01-256695

Communication establishing method between objects in different address spaces, involves generating proxy object relative to target object and using proxy object for establishing communication between two objects

Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)
Inventor: BANDA V; CAMPAGNONI F R; **CONNER M H ; COPELAND G P** ; SHEPLER E E; SMITH M G
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 6226690	B1	20010501	US 9377219	A	19930614	200137	B

Priority Applications (No Type Date): US 9377219 A 19930614
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 6226690	B1		18	G06F-015/16	

Abstract (Basic): US 6226690 B1

NOVELTY - The need for a communication link between the client process and a target object which is located in a server process's address space is determined during runtime of process in client (300). A proxy object corresponding to the target object, which emulates the target object, is generated in the client's address space. The link between the client process and the target object is established through the proxy object.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:

(a) Communication establishing apparatus;

(b) Data processing system;

(c) Communication establishing program

USE - In data processing systems for providing communication between object which are in different address spaces.

ADVANTAGE - Establishing effective communication between objects which are in different address space during runtime, is achieved, by generating proxy object for target object, during runtime and establishing communication between the process and the target object via the proxy object.

DESCRIPTION OF DRAWING(S) - The figure shows the block diagram illustrating client process bootstrapping communication with server process and activating, invoking and calling object located in server process.

Client (300)

pp; 18 DwgNo 5B/18

Title Terms: COMMUNICATE; ESTABLISH; METHOD; OBJECT; ADDRESS; SPACE; GENERATE; OBJECT; RELATIVE; TARGET; OBJECT; OBJECT; ESTABLISH; COMMUNICATE; TWO; OBJECT

Derwent Class: T01

International Patent Class (Main): **G06F-015/16**

File Segment: EPI

6/5/21 (Item 21 from file: 350)

DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

011173276 **Image available**

WPI Acc No: 1997-151201/199714

XRPX Acc No: N97-125011

Demand transmission for data processing system - by establishing communication from demand person body to target body after external body request written in body is formed in first address space

Patent Assignee: IBM CORP (IBMC); INT BUSINESS MACHINES CORP (IBMC)

Inventor: CAMPAGNONI F R; **CONNER M H** ; HUDLI R V; SMITH M G

Number of Countries: 002 Number of Patents: 003

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
JP 9026879	A	19970128	JP 95288874	A	19951107	199714	B
JP 3072709	B2	20000807	JP 95288874	A	19951107	200042	
US 6182154	B1	20010130	US 94342508	A	19941121	200108	
			US 97838458	A	19970407		

Priority Applications (No Type Date): US 94342508 A 19941121; US 97838458 A 19970407

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
JP 9026879	A		25	G06F-009/44	
JP 3072709	B2		23	G06F-009/44	Previous Publ. patent JP 9026879
US 6182154	B1			G06F-009/00	Cont of application US 94342508

Abstract (Basic): JP 9026879 A

The method involves transmitting a demand to the target body of a second address space from the demand person of a first address space. A body request that processes the demand to a remote body is passed. A distinction that the demand is outside the body request is required to the target body in response.

The external body request in which the target body is written, is searched. The external body request written in the body is formed in the first address space. A communication from the demand person body to the target body is established.

ADVANTAGE - Obtains small effect to ORB enforcement and offers all support required for dynamic prodn. of body.

Dwg.3/13

Title Terms: DEMAND; TRANSMISSION; DATA; PROCESS; SYSTEM; ESTABLISH; COMMUNICATE; DEMAND; PERSON; BODY; TARGET; BODY; AFTER; EXTERNAL; BODY; REQUEST; WRITING; BODY; FORMING; FIRST; ADDRESS; SPACE
Derwent Class: T01
International Patent Class (Main): **G06F-009/00 ; G06F-009/44**
International Patent Class (Additional): **G06F-009/46 ; G06F-012/00 ; G06F-013/00 ; G06F-015/16 ; G06F-015/163**
File Segment: EPI

6/5/22 (Item 22 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

010655900 **Image available**
WPI Acc No: 1996-152853/199616
XRPX Acc No: N96-128387

Minimising message traffic in transaction processing system - registering subordinate resources dynamically and avoids adding them to commit tree of root coordinator unless they have recoverable resources

Patent Assignee: INT BUSINESS MACHINES CORP (IBMC); IBM CORP (IBMC)
Inventor: COBB E E; HOLDSWORTH S A J; HOUSTON I S C; SMITH S A; **COPELAND G P**
Number of Countries: 005 Number of Patents: 007
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
EP 702295	A1	19960320	EP 95305844	A	19950822	199616 B
JP 8161212	A	19960621	JP 95220199	A	19950829	199635
JP 3048894	B2	20000605	JP 95220199	A	19950829	200032
US 6205464	B1	20010320	US 94307212	A	19940916	200118
US 6513056	B1	20030128	US 94307212	A	19940916	200311
			US 98172428	A	19981014	
EP 702295	B1	20030409	EP 95305844	A	19950822	200325
DE 69530252	E	20030515	DE 630252	A	19950822	200340
			EP 95305844	A	19950822	

Priority Applications (No Type Date): US 94307212 A 19940916; US 98172428 A 19981014
Cited Patents: 03Jnl.Ref; EP 457116
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
EP 702295	A1	E	10	G06F-009/46	
Designated States (Regional): DE FR GB					
JP 8161212	A		9	G06F-012/00	
JP 3048894	B2		10	G06F-012/00	Previous Publ. patent JP 8161212
US 6205464	B1			G06F-009/00	
US 6513056	B1			G06F-009/00	Cont of application US 94307212 Cont of patent US 6205464
EP 702295	B1	E		G06F-009/46	
Designated States (Regional): DE FR GB					
DE 69530252	E			G06F-009/46	Based on patent EP 702295

Abstract (Basic): EP 702295 A

The method involves receiving a transaction request to modify one or more of several resources. The transaction is assigned a global identifier. A coordinator which controls modification of the resources is created. The request is imported to several subordinate transaction manager domains.

This importing is done by creating a subordinate coordinator for each of the domains and encapsulating the global identifier. The subordinate coordinators are dynamically registered with the first coordinator only when the subordinate coordinator resources controlled by coordinator are modified by a transaction.

USE/ADVANTAGE - For bank automated teller machines. Optimises message traffic due to sending messages regardless of number of recoverable resources controlled by coordinator. Uses single set of logic for local and remote resources. Registers objects dynamically.

Dwg.6/6
Title Terms: MINIMISE; MESSAGE; TRAFFIC; TRANSACTION; PROCESS; SYSTEM; REGISTER; SUBORDINATE; RESOURCE; DYNAMIC; AVOID; ADD; COMMIT; TREE; ROOT; COORDINATE; RECOVER; RESOURCE
Derwent Class: T01
International Patent Class (Main): **G06F-009/00 ; G06F-009/46 ; G06F-012/00**
International Patent Class (Additional): **G06F-015/00**
File Segment: EPI

6/5/23 (Item 23 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

010382150 **Image available**
WPI Acc No: 1995-283464/199537
XRPX Acc No: N95-215779
Shared nothing transaction processing computer database system - has transaction processors connected by network with partition replicas stored on processors allowing page modification using Write-Ahead Log
Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)
Inventor: BHIDE A K; **COPELAND G P** ; GOYAL A; HSIAO H; JHINGRAN A D; MOHAN C
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
US 5440727	A	19950808	US 91809354	A	19911218	199537 B
			US 94276130	A	19940715	

Priority Applications (No Type Date): US 91809354 A 19911218; US 94276130 A 19940715
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 5440727	A		10	G06F-011/00	Cont of application US 91809354

Abstract (Basic): US 5440727 A

The system has numerous transaction processors with their own storage, connected by a network. The system has at least one partition of data sub-partitioned into pages. The partition includes primary and secondary replicas of the partition stored in two of the processors. A Write-Ahead Log protocol is used where a modification to a page in the partition is not made until stored in a non-volatile log.

A circuit generates a response indicating completion of a requested transaction on the primary replica. The secondary replica is updated by sending pages from the primary replica independently of the response. The pages of the partition which have been modified in the primary but not secondary replicas are monitored. The processors storing partition replicas can access the log.

ADVANTAGE - Provides failure safety and high availability. Provides efficient maintenance of replicas. Avoids need for special purpose hardware. Maintains replicas on separate systems with minimal extra workload. Avoids delays in either system. Avoids restricting updates to take place before or at transaction commit time. Requires minimal processing and disk accesses at secondary replica. Gives efficient recovery of database which maintains replicas when replica becomes unavailable. Minimises impact of maintaining replicas.

Dwg.2/7
Title Terms: SHARE; TRANSACTION; PROCESS; COMPUTER; DATABASE; SYSTEM; TRANSACTION; PROCESSOR; CONNECT; NETWORK; PARTITION; REPLICA; STORAGE; PROCESSOR; ALLOW; PAGE; MODIFIED; WRITING; AHEAD; LOG
Derwent Class: T01
International Patent Class (Main): **G06F-011/00**
International Patent Class (Additional): **G06F-015/00**
File Segment: EPI

6/5/24 (Item 24 from file: 350)

DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

009496518 **Image available**
WPI Acc No: 1993-190054/199324
XRPX Acc No: N93-146065

Language neutral objects in object oriented data processing system - parses and compiles information to generate bindings file which is input with method information to target language compiler to create object file which is link edited

Patent Assignee: INT BUSINESS MACHINES CORP (IBMC); IBM CORP (IBMC)
Inventor: **CONNER M H** ; MARTIN A R; RAPER L K; RAPER A R M L K
Number of Countries: 013 Number of Patents: 010
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
EP 546683	A2	19930616	EP 92310241	A	19921109	199324	B
CA 2077273	A	19930613	CA 2077273	A	19920901	199336	
CN 1073276	A	19930616	CN 92113213	A	19921123	199413	
EP 546683	A3	19931201	EP 92310241	A	19921109	199513	
US 5428792	A	19950627	US 91805668	A	19911212	199531	
			US 94329798	A	19941026		
CA 2077273	C	19961203	CA 2077273	A	19920901	199708	
SG 44356	A1	19971219	SG 952206	A	19921109	199808	
KR 9708530	B1	19970524	KR 9223713	A	19921209	199943	
EP 546683	B1	20000119	EP 92310241	A	19921109	200009	
DE 69230578	E	20000224	DE 630578	A	19921109	200017	
			EP 92310241	A	19921109		

Priority Applications (No Type Date): US 91805668 A 19911212; US 94329798 A 19941026
Cited Patents: No-SR.Pub; 2.Jnl.Ref; US 4667290
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
EP 546683	A2	E	41	G06F-009/44	
Designated States (Regional): CH DE FR GB IT LI NL SE					
EP 546683	B1	E		G06F-009/44	
Designated States (Regional): CH DE FR GB IT LI NL SE					
DE 69230578	E			G06F-009/44	Based on patent EP 546683
US 5428792	A		27	G06F-009/45	Cont of application US 91805668
CA 2077273	A			G06F-009/45	
CN 1073276	A			G06F-009/06	
EP 546683	A3			G06F-009/44	
CA 2077273	C			G06F-009/45	
SG 44356	A1			G06F-009/44	
KR 9708530	B1			G06F-015/16	

Abstract (Basic): EP 546683 A

The object oriented data processing system generates an object loadable module from a language neutral source program. The language neutral source program is stored and processed by a precompiler operable in a first phase to produce intermediate output files containing binary class information.

A language emitter generates language bindings which include a class data structure in a second phase. The language bindings are compiled to provide the object loadable module. Target languages include C, Fortran, C + +, COBOL or any compiled language whether or not the particular language has object programming support. Messages are displayed on a display to aid a user.

ADVANTAGE - Redefines language dependent object definitions as neutral set of information from which object support for any language, including support between languages, is available.

Dwg.4/15

Title Terms: LANGUAGE; NEUTRAL; OBJECT; OBJECT; ORIENT; DATA; PROCESS; SYSTEM; COMPILE; INFORMATION; GENERATE; BIND; FILE; INPUT; METHOD; INFORMATION; TARGET; LANGUAGE; COMPILE; OBJECT; FILE; LINK; EDIT
Derwent Class: T01
International Patent Class (Main): **G06F-009/06 ; G06F-009/44 ; G06F-009/45 ; G06F-015/16**

International Patent Class (Additional): **G06F-009/40**
File Segment: EPI

```
Set     Items   Description
S1       1354   AU=(COPELAND, G? OR COPELAND G? OR CONNER, M? OR CONNER M?
            OR FLURRY, G? OR FLURRY G?)
S2          0   S1 AND COMMAND()CACHE
S3          0   S1 AND COFETCHING
File   2:INSPEC 1969-2004/Jul W4
         (c) 2004 Institution of Electrical Engineers
File   6:NTIS 1964-2004/Aug W1
         (c) 2004 NTIS, Intl Cpyrght All Rights Res
File   8:Ei Compendex(R) 1970-2004/Jul W4
         (c) 2004 Elsevier Eng.  Info. Inc.
File  34:SciSearch(R) Cited Ref Sci 1990-2004/Jul W4
         (c) 2004 Inst for Sci Info
File  35:Dissertation Abs Online 1861-2004/May
         (c) 2004 ProQuest Info&Learning
File  65:Inside Conferences 1993-2004/Aug W1
         (c) 2004 BLDSC all rts. reserv.
File  92:IHS Intl.Stds.& Specs. 1999/Nov
         (c) 1999 Information Handling Services
File  94:JICST-EPlus 1985-2004/Jul W2
         (c)2004 Japan Science and Tech Corp(JST)
File  95:TEME-Technology & Management 1989-2004/Jun W1
         (c) 2004 FIZ TECHNIK
File  99:Wilson Appl. Sci & Tech Abs 1983-2004/Jul
         (c) 2004 The HW Wilson Co.
File 103:Energy SciTec 1974-2004/Jul B2
         (c) 2004 Contains copyrighted material
File 144:Pascal 1973-2004/Jul W4
         (c) 2004 INIST/CNRS
File 202:Info. Sci. & Tech. Abs. 1966-2004/Jul 12
         (c) 2004 EBSCO Publishing
File 233:Internet & Personal Comp. Abs. 1981-2003/Sep
         (c) 2003 EBSCO Pub.
File 239:Mathsci 1940-2004/Sep
         (c) 2004 American Mathematical Society
File 275:Gale Group Computer DB(TM) 1983-2004/Aug 04
         (c) 2004 The Gale Group
File 434:SciSearch(R) Cited Ref Sci 1974-1989/Dec
         (c) 1998 Inst for Sci Info
File 647:CMP  Computer Fulltext 1988-2004/Jul W4
         (c) 2004 CMP Media, LLC
File 674:Computer News Fulltext 1989-2004/Jul W4
         (c) 2004 IDG Communications
File 696:DIALOG Telecom. Newsletters 1995-2004/Jul 23
         (c) 2004 The Dialog Corp.
```